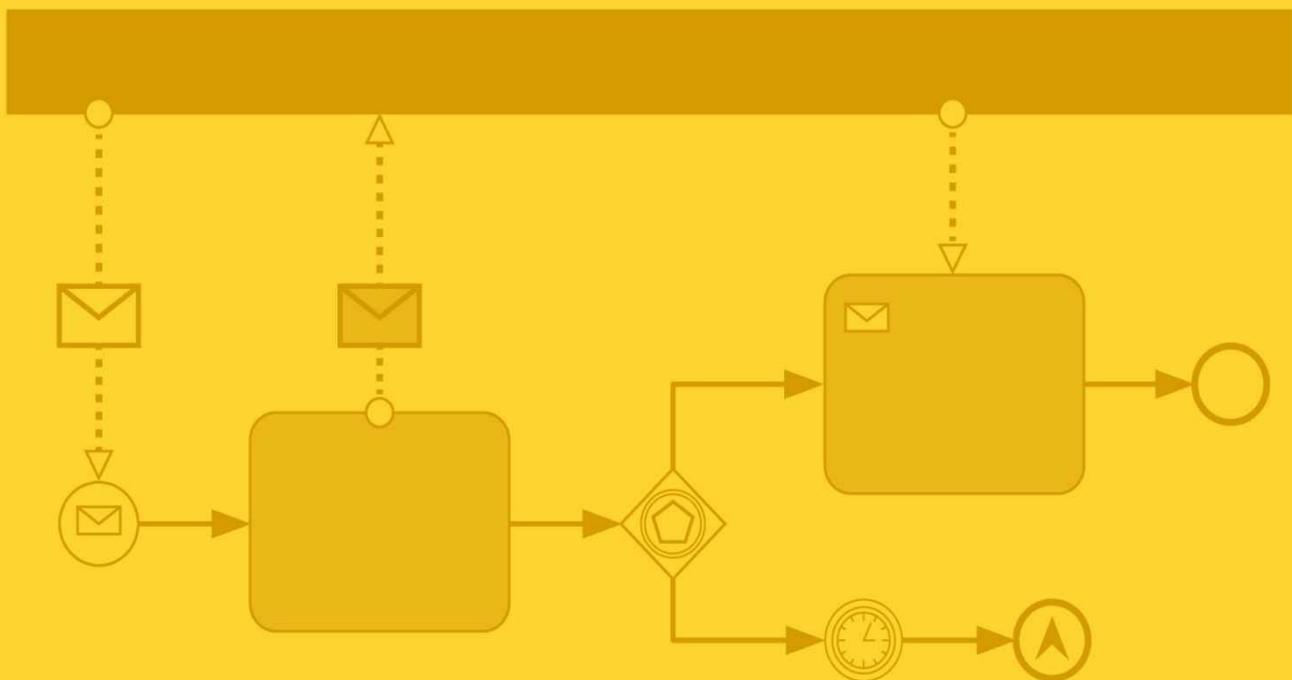


Jakob **FREUND**
Bernd **RÜCKER**
Bernhard **HITPASS**

BPMN 2.0

Manual de Referencia y Guía Práctica

4^a Edición actualizada



BPMN 2.0

Manual de Referencia y Guía Práctica

Jakob Freund Bernd Rücker Bernhard Hitpass
Cuarta Edición 2014

BPMN 2.0 Manual de Referencia y Guía Práctica Todos los derechos reservados
Prof. Bernhard Hitpass
Dept. Informática
Universidad Técnica Federico Santa María

Santiago de Chile

Edición hispana
Cuarta edición actualizada Marzo 2014

Todas las marcas y nombres mencionados en este libro son marcas registradas o marcas de sus respectivas organizaciones. Cualquier omisión o mal uso no debe ser interpretado como un intento de infringir la propiedad de otros. La publicación reconoce y respeta todas las marcas usadas por las organizaciones como un instrumento para distinguir sus derechos de productos.

Ninguna parte de este libro puede ser reproducida o usada de ninguna forma, gráfica, electrónica, fotocopiado, sin el permiso expreso y por escrito del autor, excepto en los casos de breves reseñas y referencias en artículos, publicaciones científicas y conferencias.

BPMCenter (www.bpmcenter.cl)

Layout: Autores con L^ATEX

Diseño carátula: Felipe Espinoza Guerraty, ZAGUE Diseño Encuadernación
HotMelt. + Costura Hilo-cod142201

Impresión: Empresas Dimacofi - Santiago de Chile

Copyright © (RPI Chile): N° 201720 Edición Internacional:

ISBN-13: 978-1460903933

Edición Chile:

ISBN: 978-956-345-182-5

Presentación

Uno de los ejes principales del avance de la informática, como disciplina y como ingeniería, ha sido identificar y sistematizar mayores niveles de abstracción: las funciones han dado paso a los programas, los módulos, los objetos, los servicios, los aspectos, y finalmente (por ahora) los procesos que requieren y justifican estos elementos.

Históricamente, estos niveles de creciente abstracción han sido mayoritariamente identificados en la praxis profesional ("in the wild", salvajes, por así decir) y sistematizados en la academia (domando, describiendo, sistematizando). El largo baile praxis-academia ha sido muy fructífero para nuestra profesión, y todo indica que lo seguirá siendo en el futuro predecible.

No es por lo tanto sorpresa que sea precisamente Bernhard Hitpass, con vasta experiencia industrial en las disciplinas de procesos y ahora un activo miembro de la comunidad académica, quien sea un puente no sólo entre industria y academia, sino entre el mundo iberoamericano y el germánico, hoy por hoy metodológicamente más riguroso y avanzado en estas disciplinas que otros polos comparables. Y no sólo nos trae un mensaje, sino que lo enriquece con aportes propios significativos.

Ha sido un placer colaborar estos años con Bernhard en la UTFSM, y estoy cierto de que este libro será una referencia permanente de la disciplina en nuestro idioma.

Prof. Dr. Hernán Astudillo

Universidad Técnica Federico Santa María (UTFSM) Febrero 2011 - Chile

Prólogo

El lector que comience a leer o estudiar este trabajo se encuentra con la cuarta edición de este libro. BPMN es un estándar administrado por la OMG (Object Management Group) y ahora se ha convertido además a partir de mediados del año 2013 en una norma para el modelamiento e implementación de procesos definida por la ISO (International Organization for Standardization) que lleva el código ISO / IEC 19510:2013.

Con este hecho se refuerza aún más la importancia que tiene para las organizaciones adoptar este estándar y preguntarse si tiene sentido seguir utilizando otras técnicas y notaciones para levantar, modelar, especificar e implementar procesos de negocio. Una reciente encuesta a nuestra comunidad de profesionales del BPM Center en LinkedIn nos muestra, que actualmente mas del 60 % de las organizaciones están utilizando BPMN como notación y el 7 % de esta comunidad que utiliza otras técnicas se están o se van a cambiar a BPMN en corto plazo, llegando así a un 67 % de adopción a la fecha (ver figura 1). Es muy probable que en dos años más esta cifra esté sobre el 80 % incluso bordeando los 90 %.

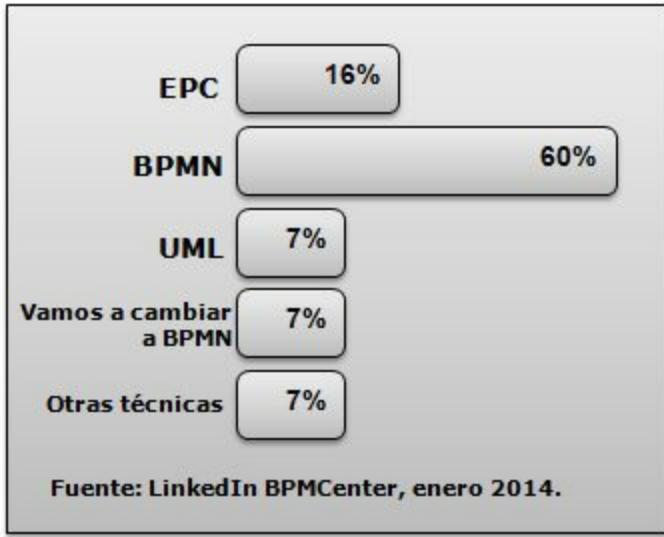


Figura 1: Popularidad de notaciones para modelar procesos

Este libro fue publicado en su primera versión en alemán en el año 2009 por los autores Jakob Freund y Bernd Rücker, reconocidos expertos de BPM en Alemania y colaboradores del estándar BPMN en el instituto de normalización internacional OMG (Object Management Group). Los autores no sólo estudiaron a fondo la notación y su aplicabilidad, sino que además la pusieron a prueba implementando los procesos en numerosos proyectos reales con el lenguaje oficial BPMN 2.0 XML y utilizando en gran parte una plataforma (propia) que interpreta este lenguaje gráfico. En el año 2013 se tomó la decisión de lanzar esta plataforma a la comunidad de desarrolladores de Java como Process Engine BPMN 2.0 en la modalidad open source (<http://camunda.org>). La iniciativa despertó tanto interés, que a la primera reunión de usuarios sobre la plataforma de BPM, asistieron más de 700 personas.

Este trabajo no sólo se limita a explicar la notación, sino que además presenta una guía práctica basada en buenas prácticas y deducidas de proyectos reales de BPM. En este sentido el trabajo fue estructurado en dos grandes partes, explicación de la notación en detalle y una guía práctica que cubre los tres niveles de abstracción, nivel descriptivo (ejecutivo), nivel operacional (negocio) y nivel técnico (ejecución técnica).

El libro al inicio tuvo tanto éxito que la primera edición se agotó a los seis meses. A mediados del 2010 la «Finalization Task Force (FTF)» de la OMG había anunciado la finalización del trabajo para que los gremios pertinentes de la OMG puedan liberar la nueva versión 2.0, de forma que la segunda edición en 2010 incorporó completamente la nueva versión al libro [FreRue10]. Entretanto mientras se escribía la primera versión hispana de este libro, en enero del 2011 la OMG oficializó la tan esperada nueva versión BPMN 2.0, un documento de más de 500 páginas que el lector la puede bajar directamente del sitio de la organización [Obj11].

A mediados del año 2010 los autores acordaron una colaboración en materias

BPMN para la lengua española con el Prof. Bernhard Hitpass, quién actualmente dirige el BPM Center ligado a la Universidad Técnica Federico Santa María en Chile. En este sentido el trabajo representa en su esencia una traducción del libro original de BPMN en alemán, pero también incluye algunas ampliaciones, como son la introducción a BPM (1.1.1), una reseña histórica de las técnicas de modelamiento (2.13.1), clasificación de técnicas de modelamiento de procesos (2.13.2) y el capítulo 6 Técnicas de Análisis y Mejora. También se hicieron algunas pequeñas ampliaciones y/o modificaciones conceptuales de común acuerdo con los autores. Principalmente existe una pequeña variación en la representación del ciclo BPM y la sección 5.3 Automatización de Procesos con BPMN 2.0, presenta una versión resumida del texto original en alemán. Por otro lado se adaptaron a la región hispana algunos párrafos sobre expresiones idiomáticas, ejemplos, indicaciones sobre asociaciones y otras materias que son propias de los países de habla germana.

Con mucho agrado podemos comunicar a los lectores que a nivel internacional hasta fines del año 2013 se han pedido miles de ejemplares. Actualmente las versiones de los libros se encuentran en el siguiente estado:

La versión alemana publicará a mediados del 2014 su cuarta edición actualizada
La versión en inglés fue publicada en el año 2012 [FreundRuecker12] y será actualizada como segunda versión a fines del 2014

La versión en español se encuentra con esta publicación en su cuarta edición

Notas generales para el lector:

La versión de BPMN 2.0.1 no ha sido modificada desde su fecha de aprobación en enero del 2011 por la OMG, razón por la cual no es necesario actualizar la descripción de la notación hasta la fecha en este libro.

En la cuarta edición se ha modificado la estructura de notas con respecto a las versiones anteriores, porque la estructura tradicional de «notas al pie» no es compatible con los formatos para «eBooks». Los eBooks tienen que adaptarse a diferentes formatos de lectura, por consiguiente se elimina el concepto de notas al pie y éstas fueron incorporadas al texto o bien integradas como un «marco de nota» en el párrafo correspondiente.

En la sección 2.12 Modelos de Coreografía, se incorporó una breve descripción de un caso real en el rubro de logística de un modelo de coreografía, resultado de un trabajo de investigación presentado en la JCCC 2012, [HitpassLeiva12]

Bernhard Hitpass, Director Ejecutivo BPM Center, Depto Informática, Universidad Técnica Federico Santa María, Santiago, Chile, email: bernhard.hitpass@usm.cl Jakob Freund, Socio Director Camunda Services GmbH, Berlin, Alemania, email: jakob.freund@camunda.com

Bernd Rücker, Socio Director Camunda Services GmbH, Berlin, Alemania, email: bernd.ruecker@camunda.com

Agradecimientos

Este libro no habría sido posible sin la colaboración de muchos profesionales y académicos, es decir lo podríamos haber escrito pero habría resultado algo horrible. Muy agradecido estamos del Prof. Dr. Thomas Allweyer [All08]también autor de una obra que introduce a BPMN. Valoramos su apoyo por habernos dado un feedback sobre nuestros textos y conceptos.

Junto a Gero Decker, Alexander Großkopf, Prof. Dr. Jan Mendling, Dr. Frank Puhlmann, Torben Schreiter y Matthias Weidlich los autores fundaron el círculo «Berliner BPM-Offensive (bpmb.de)». Todos ellos son expertos de BPMN y gracias a su colaboración pudimos corregir errores y contradicciones en el manuscrito. Dr. Frank Michael Kraft es un especialista en el diseño técnico de BPMN y nos apoyó muchísimo en el desarrollo del capítulo 5.

Thomas Niebisch aportó con su gran conocimiento de Requirements Engineering. Sus ideas de acoplamiento de BPMN con UML fueron un impulso para el desarrollo del framework aquí presentado.

Para la versión hispana de este libro, quedo muy agradecido de mi alumna, Ingeniera Civil Informática y M.Sc. (UTFSM) María Jesús Faúndes, por el arduo trabajo de revisión del manuscrito.

Igualmente, quedo agradecido de mi colega Ingeniero Civil Matemático y MBA Patricio Veloz, un colaborador con mucha experiencia en análisis, simulación y costeo de actividades, por sus valiosos aportes al capítulo 6, en especial las secciones de análisis de tiempo de ciclo y análisis de costeo de actividades.

No puedo dejar de mencionar a mi colega y Subdirector de Informática UTFSM, Prof. Dr. Hernán Astudillo, por su gran apoyo en el fomento del desarrollo científico y profesional de BPM-SOA, permitiendo la creación del BPM Center en el año 2008 (perteneciente a la UTFSM), un centro de competencia dedicado exclusivamente a este tema.

Finalmente quiero agradecer a todos los lectores que me han escrito y han publicado sus evaluaciones en los medios como Google Books o Amazon y también aquellos que han contribuido a mejorar la calidad ortográfica del manuscrito; entre ellos Domingo Cruz, correcciones que fueron consideradas en la pasada edición 2013.

Índice general

1. Introducción 1

1.1. Business Process Management	1
1.1.1. Introducción y definición de BPM	1
1.1.2. BPM en las Organizaciones	4
1.1.3. Ciclo BPM	5
1.1.4. Automatización de los Procesos	8
1.2. ¿Por qué BPMN?	10
1.3. ¿Puede BPMN cerrar las brechas?	13
1.3.1. La Contradicción	13
1.3.2. Los participantes en un proceso	13
1.3.3. Un Marco Estructural para BPMN	16

2. La Notación en Detalle 19

2.1. Entender BPMN	19
2.1.1. Lo que BPMN debe cumplir y lo que no	20
2.1.2. Los elementos básicos de BPMN	20
2.1.3. Las diferentes vistas de un proceso	22
2.1.4. Modelos, instancias, token y correlaciones	23
2.1.5. BPMN en español	24
2.1.6. Simbología y atributos	25
2.2. Actividades simples y Eventos indefinidos	25
2.3. Flujo de Procesos con Gateways	27
2.3.1. Gateway exclusivo de datos (XOR)	27
2.3.2. Gateway paralelo	30
2.3.3. Gateway inclusivo de datos (OR)	34
2.3.4. El flujo por defecto	37
2.3.5. Gateway complejo	38
2.4. Flujo de Procesos sin Gateways	41

i ii ÍNDICE GENERAL

2.5. Lanes	44
2.6. Eventos	48
2.6.1. Importancia en BPMN	48
2.6.2. Evento de mensaje	53
2.6.3. Evento de tiempo	56
2.6.4. Evento de error	58
2.6.5. Evento condicional	59
2.6.6. Evento de señal	59
2.6.7. Evento de término	60

<u>2.6.8. Evento de conexión</u>	62
<u>2.6.9. Evento de compensación</u>	63
<u>2.6.10. Evento múltiple</u>	67
<u>2.6.11. Evento múltiple paralelo</u>	68
<u>2.6.12. Evento de escalación</u>	69
<u>2.6.13. Evento de cancelación</u>	69
<u>2.6.14. Evento de Gateway exclusivo basado en eventos</u>	69
<u>2.6.15. Evento de Gateway paralelo basado en eventos</u>	72
<u>2.7. Actividades Especiales</u>	73
<u>2.7.1. Tipos de actividades</u>	73
<u>2.7.2. Propiedades de actividades</u>	76
<u>2.7.3. Actividades globales</u>	79
<u>2.8. Subprocesos</u>	80
<u>2.8.1. Reducción de complejidad</u>	80
<u>2.8.2. Modularizar y reutilizar</u>	85
<u>2.8.3. Eventos sobreimpuestos que impactan sobre el subproceso</u>	87
<u>2.8.4. Propiedades de Subprocesos</u>	89
<u>2.8.5. Subproceso del tipo transaccional</u>	91
<u>2.8.6. Subproceso impulsado por eventos</u>	92
<u>2.9. Pools y Fluxos de Mensajes</u>	95
<u>2.9.1. El dirigente y su orquesta</u>	95
<u>2.9.2. Reglas de aplicación</u>	98
<u>2.9.3. El arte de la colaboración</u>	99
<u>2.9.4. Pools cerrados</u>	102
<u>2.9.5. Pool con instancias múltiples</u>	104
<u>2.10. Objetos de datos en BPMN 2.0</u>	105
<u>2.11. Artefactos</u>	108
<u>2.11.1. Comentarios y agrupación</u>	108
<u>2.11.2. Artefactos y simbología propia</u>	110
<u>2.12. Modelos de Coreografía y Conversación</u>	111
<u>2.13. BPMN comparado con otras Notaciones</u>	115

ÍNDICE GENERAL iii

<u>2.13.1. Historia de técnicas para modelado de sistemas y procesos</u>	116
<u>2.13.2. Clasificación de las técnicas de modelamiento</u>	120
<u>2.13.3. Cadenas de procesos impulsadas por eventos (eEPC)</u>	124
<u>2.13.4. UML Diagrama de actividades</u>	125
<u>2.13.5. Indicadores y probabilidades</u>	129

3. Nivel 1: Modelo de Procesos Descriptivos 131

<u>3.1. Sobre este Nivel</u>	131
<u>3.1.1. Objetivo y Beneficios</u>	131
<u>3.1.2. Requerimientos para el modelo</u>	133

<u>3.1.3. Técnicas de levantamiento en el primer nivel</u>	<u>134</u>
<u>3.2. Caso de Estudio: Proceso de Contratación de Personal</u>	<u>137</u>
<u>3.3. Restricción de la Tabla de Tipos de Objeto (Simbología)</u>	<u>139</u>
<u>3.3.1. Pool y lanes en el nivel descriptivo</u>	<u>140</u>
<u>3.3.2. Actividades y subprocessos en el nivel descriptivo</u>	<u>142</u>
<u>3.3.3. Gateways en el nivel descriptivo</u>	<u>144</u>
<u>3.3.4. Eventos y event Gateways en el nivel descriptivo</u>	<u>147</u>
<u>3.3.5. Datos y artefactos en el nivel descriptivo</u>	<u>149</u>
<u>3.3.6. Artefactos propios en el nivel descriptivo</u>	<u>150</u>
<u>3.3.7. Ocultar y mostrar objetos en el diagrama</u>	<u>152</u>
<u>3.4. Reflexiones sobre el Análisis en el Primer Nivel</u>	<u>153</u>
<u>3.5. Consideraciones: Primer Nivel y BPMN 2.0</u>	<u>157</u>

4. Nivel 2: Modelo de Procesos Operacional 161

<u>4.1. Sobre este Nivel</u>	<u>161</u>
<u>4.1.1. Objetivo y Beneficios</u>	<u>161</u>
<u>4.1.2. Requerimientos para el modelo</u>	<u>163</u>
<u>4.1.3. Procedimiento</u>	<u>164</u>
<u>4.2. Del Nivel 1 al Nivel 2</u>	<u>166</u>
<u>4.3. Procesos de los Participantes</u>	<u>169</u>
<u>4.4. Preparación de la Automatización de los Procesos</u>	<u>174</u>
<u>4.4.1. Consideraciones al implementar procesos con un process engine</u>	<u>174</u>
<u>4.4.2. El flujo propio del process engine</u>	<u>177</u>
<u>4.4.3. Otros requerimientos para la implementación</u>	<u>180</u>
<u>4.4.4. Implementación de puntos de integración</u>	<u>181</u>
<u>4.4.5. Implementación del flujo a través de desarrollo propio</u>	<u>183</u>
<u>4.5. Recomendaciones Prácticas para el Nivel 2</u>	<u>185</u>
<u>4.5.1. Del happy path a la cruda realidad</u>	<u>185</u>
<u>4.5.2. El verdadero beneficio de los subprocessos</u>	<u>191</u>
<u>4.5.3. Los límites de la formalidad</u>	<u>192</u>
<u>4.5.4. Independizar las reglas de negocio de los procesos</u>	<u>194</u>
<u>4.6. ¿Filtros en el uso de Símbolos?</u>	<u>199</u>

5. Nivel 3: Modelo de Procesos Técnico 203

<u>5.1. Sobre este Modelo</u>	<u>203</u>
<u>5.1.1. Objetivos y Beneficios</u>	<u>203</u>
<u>5.1.2. Requerimientos para el Modelo</u>	<u>205</u>
<u>5.1.3. Forma de proceder en el nivel 3</u>	<u>205</u>
<u>5.1.4. Indicaciones para la lectura de este Capítulo</u>	<u>206</u>
<u>5.2. Fundamentos</u>	<u>206</u>
<u>5.2.1. Automatización de Procesos con un Process Engine</u>	<u>206</u>
<u>5.2.2. Ejecución de modelos de negocio - ¿Es posible?</u>	<u>208</u>
<u>5.3. Automatización de Procesos con BPMN 2.0</u>	<u>212</u>

<u>5.3.1. El modelo de procesos técnico</u>	<u>213</u>
<u>5.3.2. Modelamiento de datos y expressions</u>	<u>214</u>
<u>5.3.3. Invocación de servicios - ¿síncrono o asíncrono?</u>	<u>216</u>
<u>5.3.4. Integración de sistemas</u>	<u>217</u>
<u>5.3.5. Actividad de usuario</u>	<u>218</u>
<u>5.4. Semántica de Ejecución</u>	<u>218</u>
<u>5.4.1. Eventos de inicio e instancias de procesos</u>	<u>219</u>
<u>5.4.2. Eventos intermedios y cómo se implementan en TI</u>	<u>222</u>
<u>5.4.3. Correlación</u>	<u>225</u>
<u>5.4.4. Gateways</u>	<u>226</u>
<u>5.4.5. Finalizar una instancia de proceso</u>	<u>228</u>
<u>5.4.6. Transacciones de negocio vs. técnicas</u>	<u>231</u>
<u>5.4.7. Subprocesos</u>	<u>233</u>
<u>5.4.8. Loops e instancias múltiples</u>	<u>234</u>

6. Técnicas de Análisis y Mejora 237

<u>6.1. Introducción</u>	<u>237</u>
<u>6.2. Reingeniería, Rediseño y Mejora</u>	<u>238</u>
<u>6.3. Clasificación y Tipos de Mejora</u>	<u>241</u>
<u>6.3.1. Análisis de estructura</u>	<u>241</u>
<u>6.3.2. Análisis de tiempo de ciclo</u>	<u>244</u>
<u>6.3.3. Análisis de costeo de actividades</u>	<u>247</u>
<u>6.3.4. Análisis de responsabilidades</u>	<u>248</u>

ÍNDICE GENERAL v

7. Recomendaciones para Empezar 253

<u>7.1. Desarrolle su Estilo</u>	<u>253</u>
<u>7.2. Seleccione una Herramienta</u>	<u>254</u>
<u>7.3. Agradecimientos al lector</u>	<u>258</u>

8. BPMN Terminología Inglés-Español 259

Capítulo 1

Introducción

1.1. Business Process Management

¿Qué relación tiene BPMN con BPM? Primero tómese algo de tiempo para entender Business Process Management (BPM) y sólo así entenderá porque se desarrolló BPMN.

1.1.1. Introducción y definición de BPM

La globalización está demandando mayores exigencias, tanto a las empresas privadas como a las organizaciones públicas, en su capacidad de reacción frente a los cambios exigidos por el mercado. Estos pueden ser cambios en el tipo de demanda o cambios de regulaciones.

La capacidad que tienen las organizaciones de adaptar sus ofertas de bienes y servicios es parte fundamental del nuevo concepto de valor para los clientes. Los productos en si mismos no son lo suficientemente atractivos porque generalmente existe una sobre oferta y los elementos diferenciadores son sobre todo los servicios alrededor de estos productos. Estos desafíos incluyen el cumplimiento de regulaciones internas, externas e internacionales enfocadas en el control de calidad (trazabilidad), prevención del fraude y el cuidado del medio ambiente. Introducir procesos en las organizaciones que les permita entrar en un círculo virtuoso de mejora continua para dar cumplimiento a estas exigencias a través del tiempo, son los desafíos actuales a los que se encuentran sometidas las organizaciones.
[Hitpass09].

Mucha gente se confunde con qué es realmente BPM y no es sorprendente si consideramos que la comunidad de BPM no ha logrado ponerse de acuerdo en una definición común[Hitpass12]. Actualmente existen muchas definiciones de BPM.

1 Aunque todas ellas tienen algo en común también existen diferencias, sobre todo en el alcance. Algunos autores y expertos, en especial en Europa, restringen el BPM a una disciplina de gestión sin incluir explícitamente el apoyo de TI. Otros autores, específicamente en Norteamérica, definen BPM como el proceso hacia la automatización y operación de los procesos implícitamente con TI. ¿Podemos entonces concluir que no existe un entendimiento común sobre BPM? Antes de responder a esta pregunta debemos empezar por saber primero qué se entiende por un «proceso» en forma genérica:

«Un proceso es una concatenación lógica de actividades, a través del tiempo y lugar, impulsadas por eventos y que a través de su proceso de transformación, cumplen un determinado fin [Hitpass12].»

Esta definición contiene los principales elementos que describen un proceso:

Los eventos son ocurrencias externas que inician un proceso, ya que un proceso no se inicia por si sólo, algo tiene que ocurrir para que el proceso reaccione ante el suceso.

El proceso debe cumplir un determinado objetivo en las ciencias económicas, destinados a producir bienes y servicios.

A diferencia de los eventos, las actividades en un proceso consumen tiempo y recursos. Una actividad se puede definir como una «acción sobre un objeto», debido a que el proceso de transformación ocurre a través de las actividades en un proceso.

Las actividades en un proceso están encadenadas a través de una secuencia lógica que determinan en su conjunto las condiciones del negocio.

Estos elementos básicos describen en su conjunto los procesos y están contenidos en la mayoría de las notaciones para modelarlos, así también en BPMN. La definición es pura, no dice nada respecto para qué objetivos se levantan y modelan procesos en una organización.

El concepto de BPM como disciplina de gestión por procesos es más amplio, tiene objetivos claros y bien definidos:

Lograr o mejorar la «agilidad de negocio» en una organización. El concepto de agilidad de negocio se entiende como la capacidad que tiene una organización de adaptarse a los cambios del entorno a través de los cambios en sus procesos integrados.

Lograr mayor «eficacia». El concepto de eficacia se entiende como la capacidad que tiene una organización para lograr en mayor o menor medida los objetivos estratégicos o de negocio.

Mejorar los niveles de «eficiencia». Eficiencia es la relación entre los resultados obtenidos y los recursos utilizados, es decir el grado de productividad de un resultado. El término eficiencia está relacionado con todos los indicadores de productividad en cuanto a calidad, costos y tiempos.

Hoy en día, no basta que una organización sea sólo eficiente como lo podría haber sido en el pasado, porque si no es capaz de adaptarse ante los frecuentes cambios impulsados por la globalización no es eficaz, dicho de otra forma no logra cumplir

con los objetivos en el tiempo y calidad exigidos por los mercados. Sobre todo la agilidad de negocio ha cobrado mayor importancia en nuestros tiempos de la globalización. La empresa que pueda adaptarse más rápido a los constantes cambios en el mercado, que son además cada vez más frecuentes, tendrán mayores ventajas competitivas que aquellas que no logran adaptarse al ritmo que la globalización lo exige.

La pregunta crucial es entonces: ¿Qué instrumentos están utilizando las empresas para lograr mayor agilidad, eficacia y eficiencia? La respuesta es mayor control y eficiencia en la capacidad de cambio en sus procesos de negocio, porque a través de los procesos se crea valor para los clientes, ¿cómo hacerlo? A partir de principios de las años 90 nace la idea en los países industrializados de integrar las diferentes disciplinas de gestión corporativas directamente con la operación de los procesos. En una publicación de Smith and Fingar en el año 2002 con el título BPM Third Wave[SmiFin02], aparece por primera vez el acrónimo BPM. Académicos, profesionales y proveedores de TI captan rápidamente la importancia y el interés por BPM. La tendencia ha ido creciendo día a día y se han hecho grandes inversiones en el desarrollo de técnicas, metodologías y soluciones para BPM.

Volviendo entonces a nuestra pregunta inicial sobre si existe a nivel global un entendimiento común respecto de lo que es BPM, podemos concluir que si, aunque las definiciones de algunos autores se diferencien en ciertos aspectos.

Nuestra definición tiene un alcance amplio y abarca tanto la disciplina de gestión como la incorporación de TI para la automatización de los procesos. Definimos en forma abreviada BPM como una «Disciplina de Gestión por Procesos de Negocio y de Mejora Continua apoyada fuertemente por TI». Una definición más amplia la encontramos en la guía de referencia CBOK (Common Body of Knowledge) de la Asociación Internacional de Profesionales de BPM (ABPMP: Association of BPM Professionals)[ABPMP09]:

«Business Process Management (BPM) es un enfoque sistemático para identificar, levantar, documentar, diseñar, ejecutar, medir y controlar tanto los procesos manuales como automatizados, con la finalidad de lograr a través de sus resultados en forma consistente los objetivos de negocio que se encuentran alineados con la estrategia de la organización. BPM abarca el apoyo creciente de TI con el objetivo de mejorar, innovar y gestionar los procesos de principio a fin, que determinan los resultados de negocio, crean valor para el cliente y posibilitan el logro de los objetivos de negocio con mayor agilidad.»

Como lo indica la definición de la ABPMP, BPM es una disciplina integradora que engloba técnicas y disciplinas, que abarca las capas de negocio y tecnología, que se comprende como un todo integrado en gestión a través de los procesos. Nos inclinamos por esta definición porque diferencia entre procesos manuales y

automatizados, pero integra ambos casos a la disciplina de BPM. Con esta definición pretendemos lograr un entendimiento común que es necesario para tener éxito al introducir BPM en una organización. Para lograr los objetivos que se persiguen en BPM es necesario sincronizar e integrar los procesos manuales, con los implementados o los que se van a automatizar, con apoyo de TI.

1.1.2. BPM en las Organizaciones

¿Cuando entra en acción BPM? Como consultores especializados en BPM nos encontramos con las siguientes situaciones de la vida real que inducen a un proyecto de BPM:

1. Procesos existentes y actuales deben ser rediseñados y/o mejorados en su rendimiento con apoyo de tecnología.
2. Levantar y documentar procesos actuales, con la finalidad de automatizarlos u otros fines como por ejemplo preparar una certificación de ISO 9000.
3. Introducir un nuevo proceso en la organización.

En la mayoría de los casos nos encontramos con la primera situación en proyectos de BPM, ya que se espera una mejora por medio de un rediseño y una posterior implementación por medio de TI.

1.1.3. Ciclo BPM

Por lo general los modelos de BPM son muy simples o muy complejos. Si son muy simples, contienen sólo procedimientos banales y sirven a lo más para presentaciones de marketing. Mientras que si son modelos muy complejos tratan de captar todas las ocurrencias y eventualidades, amarrando u obligando al usuario en un plan de trabajo demasiado intensivo, que generalmente no es aplicable en la práctica.

Por otra parte, si no contamos con ningún modelo nos faltaría una carta de navegación para orientarnos en nuestros proyectos de BPM. A continuación presentamos un modelo que representa el ciclo de BPM, ni muy simple ni muy sofisticado pero que ha dado muy buenos resultados en la práctica (ver figura 1.1).

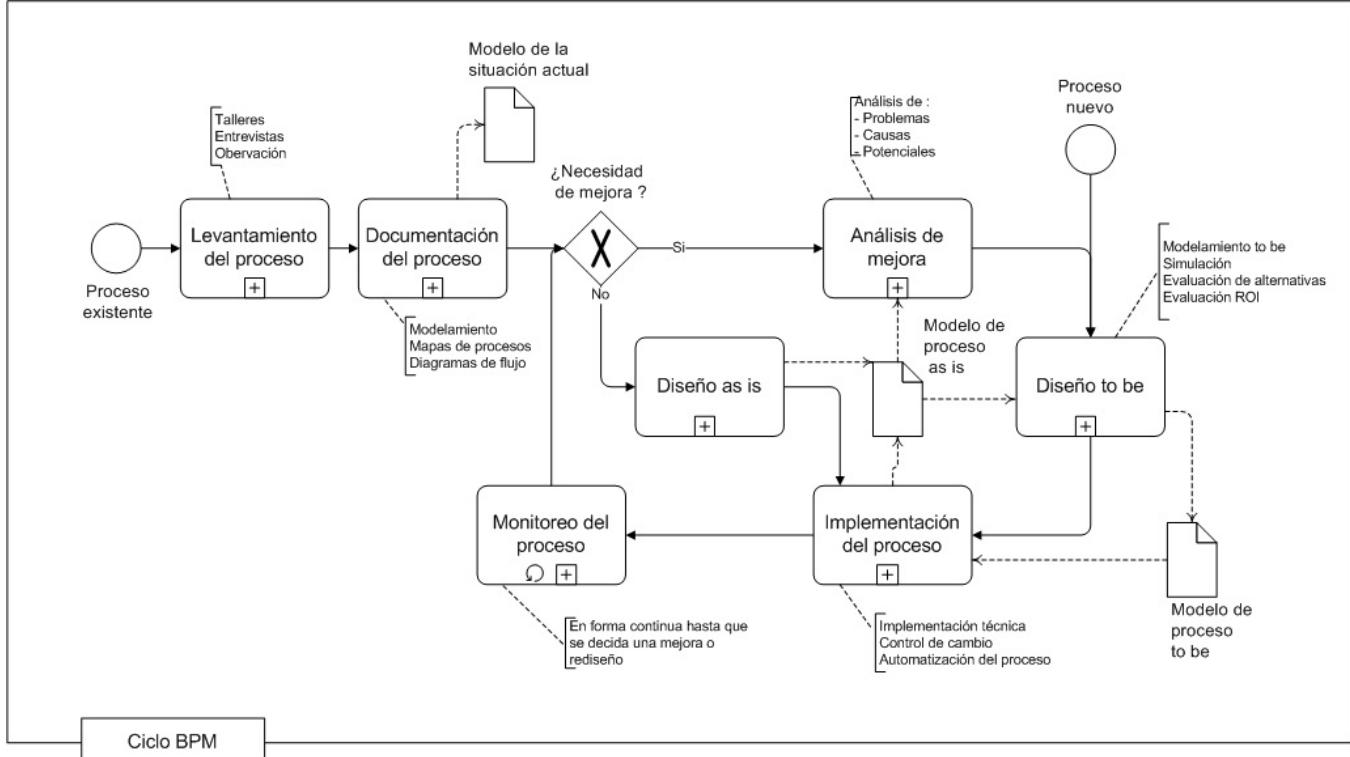


Figura 1.1: El ciclo de BPM

El ciclo está pensado para ser aplicado para cada proceso por separado o en forma independiente. Cada proceso puede encontrarse en un estado diferente del ciclo. El ciclo comienza a partir de dos posibles constelaciones:

Un proceso actual que debe levantarse y documentarse y/o rediseñarse. Se debe introducir un nuevo proceso, no existente en la organización.

En la fase de “Levantamiento del Proceso” primero se debe recoger la información sobre cómo está organizado el flujo de trabajo. Esto se realiza con la ayuda de técnicas de moderación, talleres, entrevistas, recolección de documentación, etc. Para esto en el proceso a levantar se debe:

Delimitar claramente de procesos anteriores o posteriores

Describir los servicios que produce para los clientes y qué prioridad tiene desde el punto de vista de los objetivos empresariales

Representar tanto el flujo de trabajo como los roles que intervienen en cada uno de los pasos, los recursos que se utilizan y los sistemas de información que lo apoyan

En la etapa de “Documentación del Proceso” el conocimiento adquirido en la etapa de levantamiento se documenta en un modelo de procesos que refleja la situación actual. La documentación resultante comprende los diagramas de los flujos, fichas de descripción, políticas de negocio y procedimientos que se utilizan para ejecutar el trabajo.

Las debilidades identificadas en la fase de “Análisis de mejora” o las desviaciones que muestra el “Monitoreo del Proceso” son por lo general el punto de partida para un rediseño de procesos. Eventualmente, se pueden evaluar diferentes variantes o escenarios con ayuda de simuladores. Esto aplica también si se está diseñando un proceso nuevo. En ambos casos el resultado o entregable es un modelo de procesos deseado (To be).

La etapa de “Implementación del Proceso” abarca tanto la implementación técnica como también las adaptaciones organizacionales que se requieren. La gestión del cambio (en inglés: Change Management) y la estrategia de comunicación constituyen elementos fundamentales a considerar para el éxito del proyecto. El modelo técnico puede implementarse por medio de un Process Engine o una Suite de BPM (en inglés: Business Process Management Suite, BPMS) o a través de un clásico desarrollo de software. El resultado final de la implementación técnica del proceso en la situación actual (As is) automatizado y documentado, corresponde con el modelo de proceso deseado (To be).

En la literatura y en el mercado se utilizan varios términos para sistemas que implementan procesos: sistema de workflow (WfM), Business Process Management Suite (BPMS), motor de workflow y Process Engine. De ahora en adelante vamos a utilizar preferentemente el término «Process Engine» en forma genérica que en la práctica puede ser cualquiera de ellos. Por lo general la Suite de BPM (BPMS) es el sistema más completo que trae todas las componentes integradas (modelador técnico, motor de workflow, panel de control, interfaz de usuario, APIS de integración y en algunos casos Enterprise Service Bus (ESB).

Las fases desde el “Levantamiento del Proceso” hasta la “Implementación del

Proceso” se administran por lo general por medio de la organización de un proyecto, mientras que el “Monitoreo del Proceso” (en inglés: Process controlling) se concibe como un proceso continuo y forma parte de todas las operaciones. Las actividades más importantes de “Monitoreo del Proceso” son el control constante de las operaciones (técnicamente hablamos del control de instancias de los procesos reales) y su respectiva evaluación de los indicadores. De acuerdo a la escuela de BPM, si se detectan problemas puntuales debieran corregirse de inmediato o en línea. Si hay recursos disponibles es posible solucionar problemas estructurales sin necesidad de formular un proyecto, pero si sus causas no están claras o son complejas, se hace necesario planificar e implementar un proyecto de mejora y rediseño. La decisión sobre si es necesario formular un proyecto nuevo o instalar un equipo de trabajo en operaciones, debiera tomarla el responsable del proceso de común acuerdo con los participantes.

Con esta breve explicación de cómo funciona el ciclo de BPM, el lector se dará cuenta de la importancia que tienen los modelos de procesos en BPM y junto a ello

la importancia que puede adquirir un estándar de modelamiento como BPMN. Usted puede constatar también que el modelamiento de procesos no es una etapa del ciclo de BPM, sino que es más bien una actividad transversal, porque de facto se aplica en todas las fases del ciclo, sobre todo en las fases de “Documentación del Proceso”, “Diseño As is” y “Diseño To be”. Desgraciadamente, siempre nos volvemos a encontrar con gente que confunden la “Documentación del Proceso” con el modelamiento del proceso y lo incluyen como una fase en el ciclo; esto es una equivocación.

El ciclo BPM muestra en sus principales fases cómo funciona el círculo virtuoso de mejora continua de los procesos. Para aplicarlo es necesario:

Asignar responsabilidades a los procesos y a cada uno de sus pasos
Emplear métodos de análisis y gestión en él
Contar con el apoyo de soluciones adecuadas de TI

Lograr una coordinación fluida entre estas tres componentes es tarea de gestión por procesos (BPM-Governance). Gestión por procesos se encuentra por sobre cualquier proyecto de BPM y tiene por consiguiente la misión de introducir la «Gestión por Procesos de Mejora Continua».

El primer objetivo de BPMN fue desarrollar una notación gráfica que permitiera automatizar en forma más rápida los procesos. Esta es la razón del porque usted tiene que entender los fundamentos de TI si quiere diseñar buenos modelos en BPMN. Si ha logrado entender los conceptos detrás de BPMN, habrá construido fundamentos sólidos para diseñar buenos modelos y de esta forma poder cerrar la brecha entre la capa de negocio y la capa de TI.

1.1.4. Automatización de los Procesos

¿Qué se entiende por automatización de procesos? Para entender que puede abarcar la automatización, vamos a describir primero un proceso simple de solicitud de crédito organizado en forma manual y luego vamos a describir como hoy en día se automatizan (implementan técnicamente) este tipo de procesos.

Ingresa una solicitud de crédito por correo y es derivada a un ejecutivo de negocio en el banco. El ejecutivo revisa primero la solicitud en forma visual. Luego ingresa algunos datos del solicitante en un sistema de análisis de riesgo. Si el índice de riesgo es positivo o aceptable, ingresa los datos de la solicitud en un sistema de crédito financiero y luego envía la solicitud evaluada a su superior para que la apruebe.

La automatización de este proceso podría resultar de la siguiente forma: Al arribo de la solicitud de crédito por correo, se digitaliza y vía un programa de OCR (Inglés:

Optical Character Recognition ó reconocimiento de texto automático) se extraen ciertas variables del formulario y se ingresan a un sistema de evaluación de crédito. Luego se crea un documento electrónico que gatilla la creación de una orden de trabajo en el Process Engine y es depositada en la bandeja de entrada de actividades nuevas del ejecutivo correspondiente. El ejecutivo selecciona de la lista la solicitud correspondiente y visualiza la solicitud de crédito en el Process Engine, la revisa formalmente y luego el Process Engine por medio de un servicio web invoca el sistema de análisis de riesgo enviándole la información correspondiente técnicamente se traspasan las variables). Si el resultado del análisis es positivo, el Process Engine deriva automáticamente la solicitud de aprobación a su superior, ingresando los datos en el sistema de crédito financiero por medio de un servicio web y depositándolo en la bandeja de entrada de él para su debida aprobación.

Podríamos discutir si este proceso podría ser mejorado, pero este caso describe la diferencia entre un proceso manual y uno automatizado:

Si hablamos de automatización de procesos no significa que este se encuentre completamente automatizado.

La componente central de la automatización de procesos es el Process Engine (automatización del flujo de control).

El Process Engine controla el proceso, a través del cual dirige a los usuarios que participan en las diferentes actividades y sus respectivos resultados (Human Workflow Management) y controla las interfaces internas y externas con los sistemas que participan en el proceso (Orquestación de servicios).

Las decisiones sobre qué tipo de actividades o servicios deben invocarse, las toma el Process Engine a través de la lógica técnica implementada (modelo de procesos técnico) y los puntos de intervención de los usuarios. Dicho de otra forma, no siempre la lógica del proceso implementada es mandatoria, en ciertas circunstancias puede ser influenciada por los participantes del proceso, con la salvedad que debe quedar todo registrado.

En la figura 1.2 encontrará una representación genérica de la automatización de un proceso con un Process Engine.

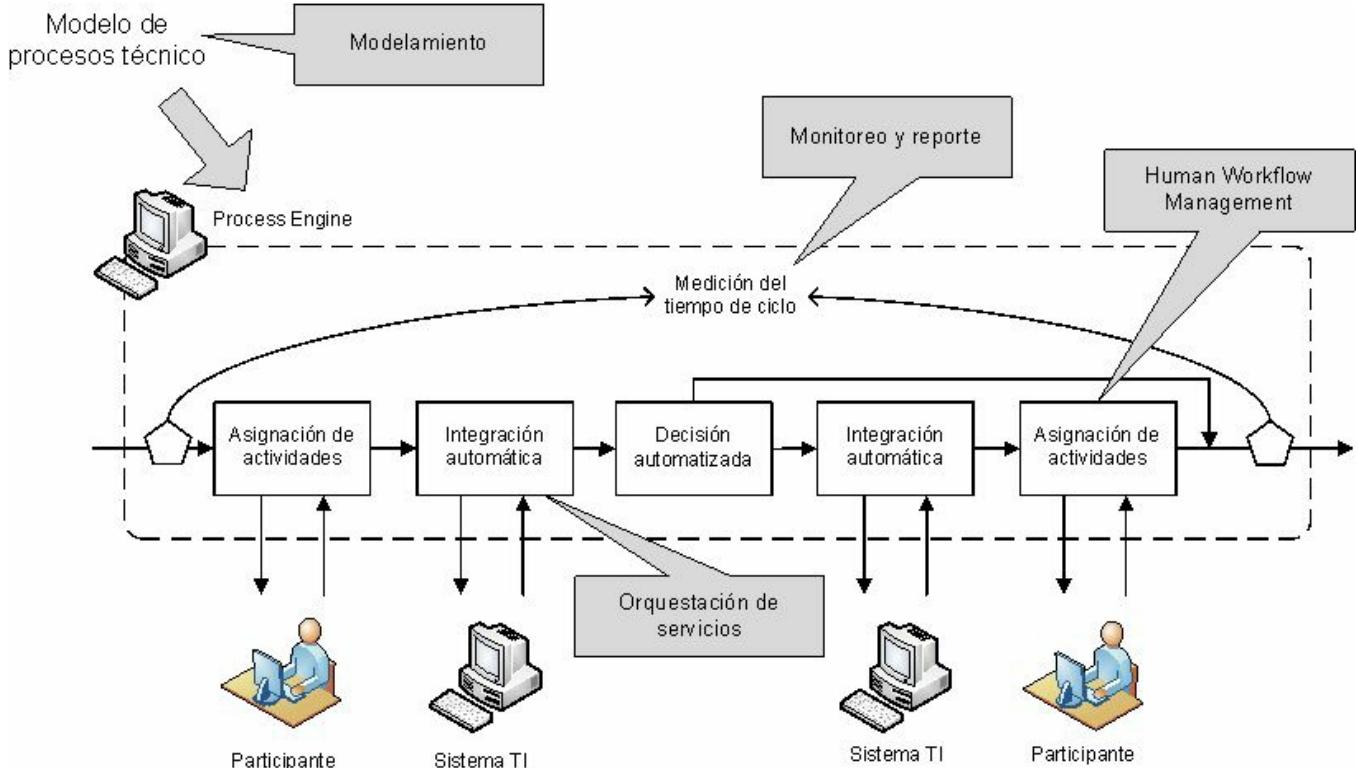


Figura 1.2: Automatización de un proceso con un Process Engine

A lo mejor el lector se imagina que un Process Engine es un entorno para desarrollar software que automatiza los procesos. En verdad es un ambiente de desarrollo pero especializado: Podríamos decir que el motor de workflow del Process Engine es el compilador o el interpretador del flujo y el modelo técnico del proceso representa el código de los programas computacionales. Pero utilizar los entornos del Process Engine tiene grandes ventajas comparado con el desarrollo propio de un workflow:

El motor de workflow se especializa en interpretar y ejecutar la lógica de los procesos y por consiguiente trae muchas funcionalidades que sin él habría que desarrollarlas y testearlas antes de poder utilizarlas. Es decir con un Process Engine un proyecto de BPM es muchísimo más productivo en la implementación de procesos. Por otro lado un Process Engine no permite representar cualquier funcionalidad (por eso es un entorno especializado) y si se requiere hay que integrarlas a un sistema.

Un Process Engine tiene la capacidad de integrar Workflow Management con soluciones de negocio (Sistemas de TI). De esta forma se convierten en plataformas técnicas poderosas que son capaces de automatizar todo tipo de procesos de principio a fin, independiente de la tecnología que hay debajo o del lugar de trabajo de los usuarios. Algunas plataformas de Process Engine incluso se complementan con un bus o orquestador de servicios, llamado Enterprise Service Bus (ESB) u otras componentes como un motor de reglas que aumentan la flexibilidad del entorno.

Como el Process Engine monitorea el proceso tiene el control absoluto sobre él, sabe en cualquier momento donde se encuentra el proceso y lleva un registro

minucioso de todo lo acaecido. De esta forma puede llevar los indicadores sobre el flujo en línea y tiempo real, que sirven como input para el monitoreo de los procesos en línea o Business Activity Monitoring (BAM). Con la existencia de un BAM se puede integrar el control de gestión sobre los procesos en tiempo real.

Estas tres funcionalidades justifican la utilización de un Process Engine, pero además existe otro beneficio muy importante: El Process Engine trabaja sobre la base de un modelo de procesos que fue desarrollado por gente de negocio. Nos encontramos ante un cambio de paradigma en el sentido de poder implementar un proceso basado en la documentación de un modelo de negocio y justamente este es el objetivo del estándar BPMN.

1.2. ¿Por qué BPMN?

En la figura 1.3 encontrará un modelo de procesos técnico, que puede ser ejecutado por un motor de procesos (Process Engine). Este modelo fue diseñado en el lenguaje Business Process Execution Language (BPEL), un estándar basado en XML para la automatización de procesos. ¿Podemos confrontar con este tipo de modelo de procesos a usuarios de negocio? ¡Con seguridad que no! Si bien es cierto que existen algunas herramientas de software que son capaces de mostrar en forma gráfica parte de estos modelos, el estándar de BPEL no cuenta con simbología propia, por lo que no poder graficar el estándar lo vuelve más técnico. Este puede 1.2. ¿POR QUÉ BPMN? 11

que sea uno de los motivos por los que hasta el día de hoy BPEL no ha logrado la aceptación esperada.

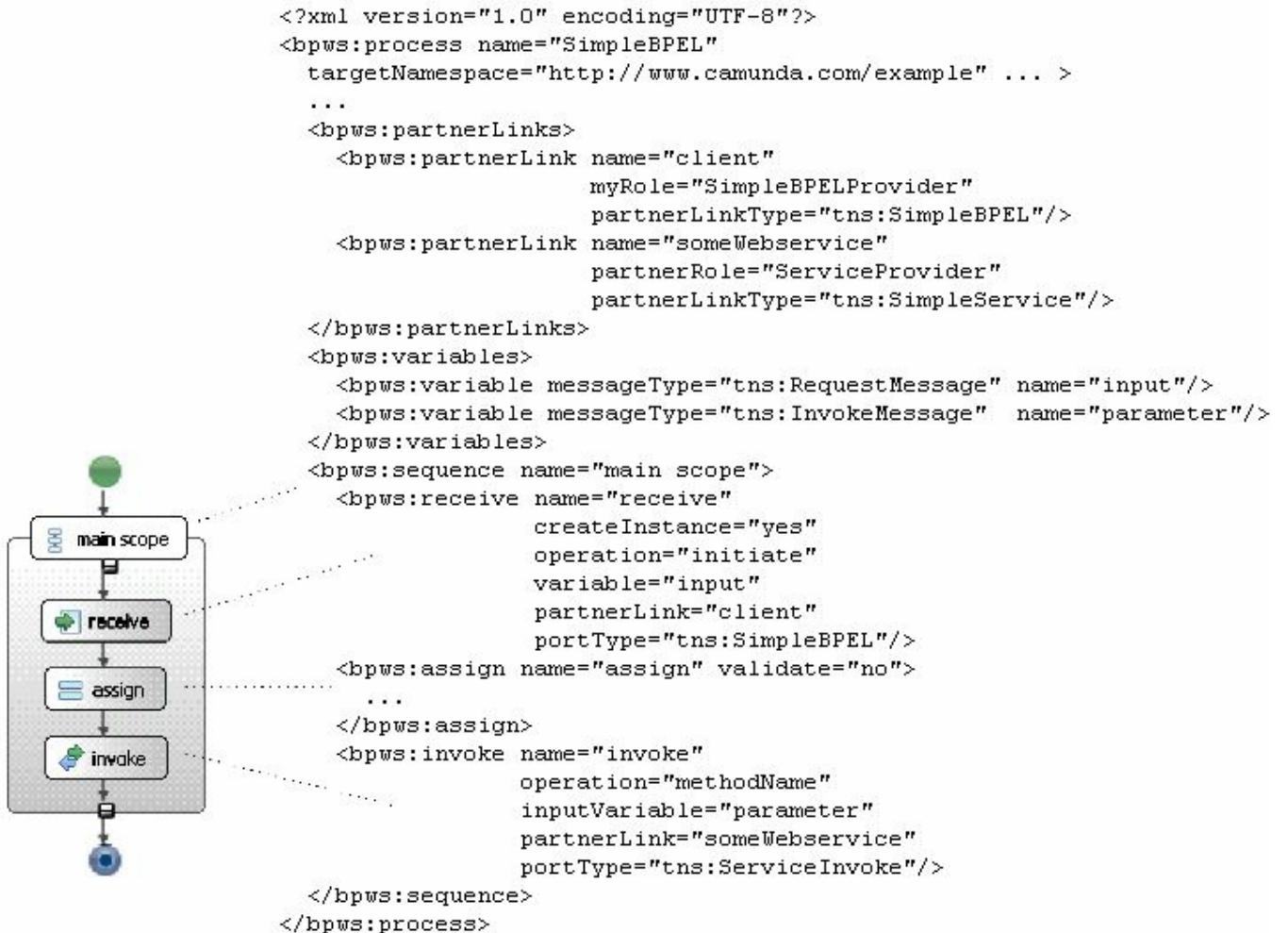


Figura 1.3: Proceso simple en BPEL en forma gráfica y como XML

La primera versión de la Business Process Modeling Notation (BPMN) fue desarrollada por el instituto Business Process Management Initiative (BPMI) principalmente bajo la tutela de Stephan A. White profesional de la IBM en 2004. Desde un principio fue el principal objetivo de disponibilizar una notación gráfica, estandarizada, que permitiera automatizar los procesos a partir del diseño gráfico. En el año 2005 fue trasladado el proyecto a la Object Management Group (OMG), debido a que el BPMI no era un instituto que administra estándares. La OMG es muy conocida en el mundo informático porque administra entre otros el estándar del lenguaje para el diseño de software llamado Unified Modeling Language (UML). A través de la OMG, de la cual son miembros la mayoría de los proveedores más importantes de TI, BPMN se difundió rápidamente a nivel mundial y casi todos los proveedores sean grandes o pequeños, académicos o consultores empezaron a adoptar este estándar.

La última versión oficial 1.2 fue publicada en enero 2009 [Obj09]. La versión 2.0, completamente nueva y ampliada, se terminó a mediados del año 2010 y a finales de este, el equipo de la OMG encargado de revisar y finalizar la nueva versión, llamada Finalization Task Force (FTF), dio la recomendación al gremio de decisión de oficializar la versión 2.0. A partir de la versión 2.0 la sigla BPMN cambia levemente de nombre a: Business Process Model and Notation.

Si usted se interesa en saber qué hay detrás o qué significa este misterioso acrónimo BPMN, se lo podemos explicar: BPMN es una especificación que se encuentra documentada en un manual no muy fácil de entender y que lo puede bajar sin costo alguno como PDF del sitio web de la OMG [Obj09]. En la versión 1.2 consta de 320 páginas y la versión 2.0 de más de 500 páginas[Obj11]. Ambas versiones sólo están disponibles en el idioma inglés. Todos los objetos, sus descripciones y sus respectivas reglas de utilización, se encuentran definidos en estos documentos.

Paradojalmente hasta la versión 1.2 no se podían mapear los modelos directamente en un entorno técnico, porque aún no estaban definidos los atributos técnicos. Debido a esta falencia existieron muchos problemas en convertir (mapear) los modelos a lenguajes de ejecución como BPEL. Recién con la versión 2.0 existe un metamodelo que permite ejecutar directamente los modelos de BPMN.

Estos dos hechos importantes de la nueva versión, es decir estandarización y habilidad de ejecución directa conlleva a los siguientes beneficios:

Para las organizaciones aumenta el grado de independencia de las herramientas de BPM, porque si cambian de herramientas no tienen que volver a capacitar en otras notaciones. Al 2010 existen más de 60 herramientas de modelación de BPMN (tendencia en aumento) y muchas de ellas las puede adquirir gratis.

La comunicación con otros socios de negocio que hayan aprendido BPMN (clientes, consultores, proveedores, etc.) será más rápida, fluida y expresiva.

Se puede esperar que nuevo personal traiga el conocimiento de BPMN.

Institutos de capacitación, universidades y empresas consultoras van a invertir recursos para formar profesionales en esta notación. Empresas privadas van a desarrollar soluciones basadas en este estándar, por ejemplo, nuestro marco metodológico propuesto en este libro (BPMN Framework) fue desarrollado para el modelamiento con este lenguaje. Pensamos que no habríamos invertido tantos recursos en desarrollar este Framework, si la notación no se hubiera convertido en un estándar.

1.3. ¿Puede BPMN cerrar las brechas? 1.3.1. La Contradicción

La notación BPMN contiene una serie de objetos representados por símbolos y a través de reglas específica como se deben relacionar estos objetos. A la definición gráfica de la simbología, su debido uso por medio de reglas bien definidas se le llama también «Syntax» o reglas sintácticas. Al significado de los símbolos y de los patrones que con ellos se pueden modelar se le llama semántica.

Sin embargo, no va a ser suficiente saber interpretar la simbología para desarrollar

«buenos» modelos de procesos.

Los autores tienen vasta experiencia desde el año 2007 en el empleo de la notación en proyectos reales.

1.3.2. Los participantes en un proceso

Si modelamos procesos debemos hacerlo pensando en los clientes finales. Dicho de otra forma, es necesario ponerse en la situación del cliente, pensando en el concepto de valor para él. Seguramente usted pensará, ¿bien por qué no? Sin embargo sólo pocos diseños de procesos cumplen con este requerimiento.

El grado de influencia y los intereses de los participantes en los procesos pueden ser muy variados y diferentes. Los roles de participantes que describimos a continuación siempre están presente de alguna forma en proyectos de BPM, pero en pocos casos nos encontramos con una organización orientada a procesos como muestra la figura 1.4. Los autores en Alemania pudieron constatar que empresas que cuentan con mayores niveles de madurez en BPM también cuentan con roles bien definidos y estructuras orientadas a procesos. De todas formas recomendamos al lector que se está iniciando en BPM, familiarizarse con los roles de los participantes en gestión orientada a procesos. Hemos mantenido los términos en inglés porque las empresas globales actúan cada vez más en mercados internacionales. El equivalente en español lo encontrará entre los paréntesis.

Process Owner (Dueño de Proceso): El process owner es el responsable de plasmar la estrategia en los procesos. El debiera tener el mayor interés de todos los participantes en promover la mejora en la eficiencia de sus procesos. En muchas ocasiones las propuestas de mejora no vienen de él, sino de otras áreas y si lo logran convencer, él disponibiliza parte o gran parte del presupuesto para un proyecto de BPM. En la mayoría de las organizaciones el process owner es miembro de la alta gerencia o es responsable de una área o línea de negocio.

Process Manager (Gestor de Procesos): El process manager es el responsable de operaciones, reporta directamente al process owner y es él quién impulsa las propuestas de mejora. Él es responsable de mantener la comunicación con los clientes y/o proveedores. Normalmente al process manager lo encontramos inserto en un nivel de jerarquía intermedia, como subdirector, subgerente, jefe de sucursal o jefe de grupo.

Process Participant (Usuario, Ejecutivo de Negocio): Los process participants son los usuarios de negocio que trabajan en operaciones con el proceso, es decir parte integrante de la cadena que crea valor para los clientes. Se pueden relacionar de muy diferentes maneras con el process manager. En la mayoría de las organizaciones son usuarios de un área funcional, como ventas, finanzas o logística. En estos casos no existe un process manager o actúe en su parte del proceso como tal y el usuario

reporta directamente al encargado del área. Si la empresa está organizada en forma matricial, lo que en empresas globales es bastante común, pueden surgir conflictos entre el process manager y los responsables de áreas. En estructuras matriciales este tipo de problemas está institucionalizado debido al conflicto de intereses que surge en el punto de intersección. Este clásico problema no lo puede solucionar el modelamiento de procesos, ni menos BPMN. La forma de introducir u orientar una estructura organizacional a los procesos, no forma parte de esta materia, por lo que recomendamos consultar la literatura de BPM al respecto.

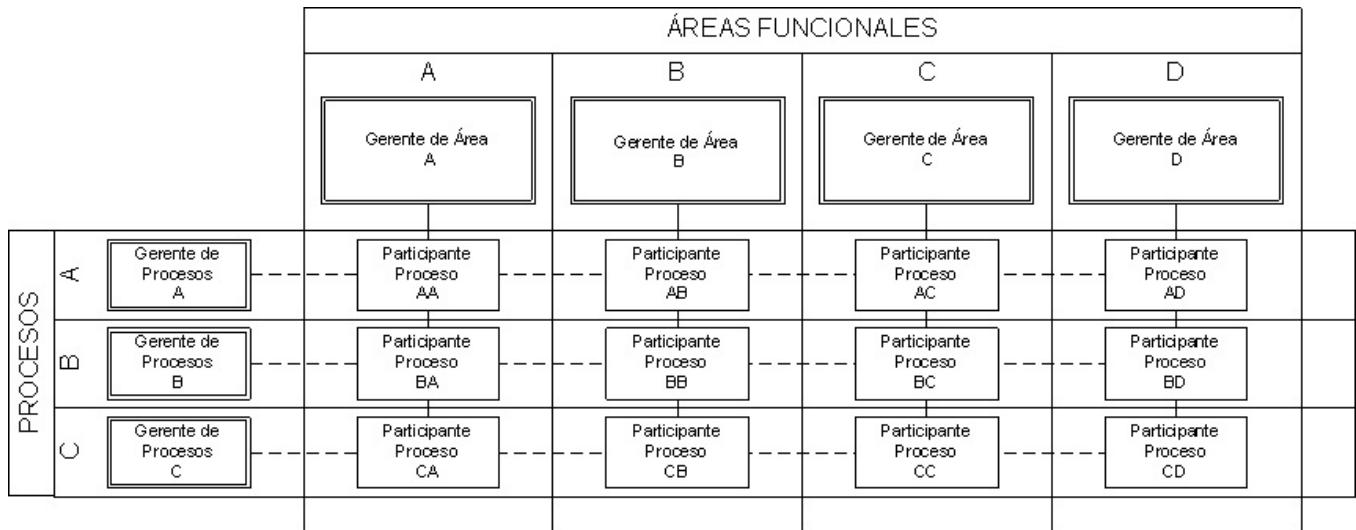


Figura 1.4: Matriz organizacional orientada a procesos

Process Analyst (Analista de Procesos): Las competencias que se esperan del analista de procesos son conocimientos de BPM en general y en este caso BPMN en específico. El analista de procesos apoya al process manager como asesor interno o externo en todas las fases del ciclo de BPM. Él puede representar como experto al process manager ante consultores externos o formar parte del equipo de proyectos de BPM. El analista de procesos puede ser miembro de un área de procesos de la empresa o pertenecer como analista al departamento de informática de la empresa. En muy pocas ocasiones será el responsable de la implementación de los procesos, a pesar que posee buenos conocimientos o una gran afinidad con las tecnologías de la información. El analista de procesos debiera de tener una gran habilidad en materias de desarrollo organizacional y técnicas de comunicación. Pero sobre todo es, como lo indica su rol, un analista. Se espera un gran dominio de la notación BPMN y como coordinador entre personas de negocio y de TI es un rol clave en cualquier proyecto de BPM. De acuerdo a nuestras observaciones y experiencias, más del 70 % de las personas que ocupan este rol no cuentan con las competencias suficientes para cumplir con este objetivo. En la mayoría de los casos porque les faltan las habilidades para este perfil. La calificación más importante de un analista de procesos no es el comunicar sino el captar o escuchar a los participantes. Buenos analistas de negocio sienten la necesidad de querer atender todo en detalle. Al mismo tiempo poseen la empatía, como para poder ponerse en el lugar del cliente y representar sus inquietudes. A ellos no se les escapa ningún detalle, pero al mismo

tiempo poseen un buen sentido de abstracción y pueden reducir los modelos a su esencia. El perfil de un jefe de proyecto es diferente, el está centrado en cumplir las metas del proyecto y por lo general prioriza las metas técnicas del proyecto, como fechas de entrega y mantención del presupuesto de costos del proyecto, ante aspectos de calidad y eficiencia. Por esta razón no se aconseja mezclar ambos roles en el sentido que un jefe de proyectos actue como analista o vice versa.

Process Engineer (Ingeniero de Procesos Técnico): El ingeniero de procesos desarrolla e implementa un modelo técnico a partir de la especificación y el diseño operacional validado por él o los analistas de procesos. El diseño técnico debiera realizarse en el mismo entorno (process engine o BPMS) en donde se implementarán los procesos. Un programador puede asumir el rol de ingeniero de procesos, si la solución será un desarrollo propio por medio de programación (Java, C# u otros lenguajes). El ingeniero de procesos también puede actuar como asesor en la fase de modelamiento de la lógica operacional.

Luego de haber discutido los roles que intervienen en un proceso a través de diferentes tipos de participantes, podemos deducir las vistas y niveles que requieren estos participantes para satisfacer sus intereses.

1.3.3. Un Marco Estructural para BPMN

En seminarios y en nuestros proyectos de consultoría hemos capacitado a muchos profesionales de muchas empresas en las prácticas de BPMN. El conocimiento que hemos adquirido como fondo de experiencia de estos proyectos lo hemos conceptualizado en un marco estructural llamado BPMN-Framework. Este framework nos permite seleccionar el tipo de objetos y patrones que utilizamos o no recomendamos de usar de acuerdo al nivel del framework.

Hasta el momento hemos tenido buenas experiencias con nuestros clientes en la aplicación de este framework. Aunque usted no lo adopte, de igual forma le puede servir como guía de orientación en sus propias convenciones de modelamiento.

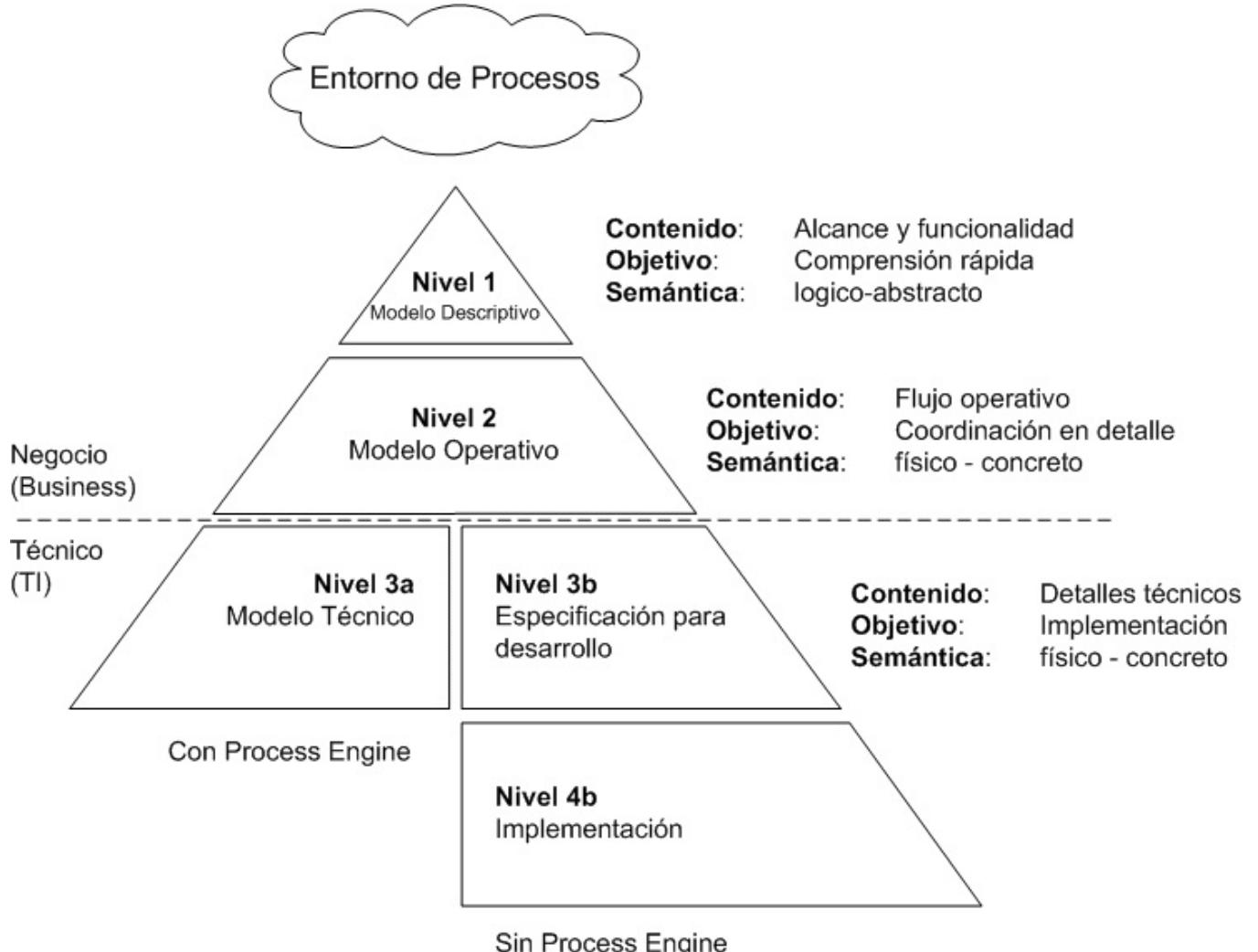


Figura 1.5: Marco estructural para BPMN (camunda BPMN-Framework)

La figura 1.5 muestra el BPMN-Framework que consta de cuatro niveles, de los cuales sólo los primeros tres son realmente relevantes para BPM y serán tratados en este libro.

Out of Scope - Entorno de Procesos: El framework fue desarrollado desde la perspectiva de un proyecto y se refiere siempre a un proceso o a un grupo de procesos (proceso de negocio) que están relacionados entre si. El modelamiento de mapas de procesos no forma parte de nuestro estudio o de este marco estructural. La notación de BPMN tampoco abarca o contempla el desarrollo de mapas de proceso, lo que queda reservado para otros marcos metodológicos de un concepto de arquitectura empresarial. Esto no impide la integración de otras notaciones como los diagramas de la cadena de valor o Value Added Chain (VAC) con diagramas de BPMN en un marco de arquitectura empresarial.

Nivel 1 - Procesos Descriptivos: La audiencia del nivel descriptivo son principalmente los Process Owner y Process Manager como también Process Participants y Process Analysts en las primeras fases de un proyecto. En este nivel queremos definir el contexto de los procesos que se deben levantar, modelar, documentar y eventualmente rediseñar. El objetivo de este nivel es además validar el

alcance y la funcionalidad principal de los procesos que deben levantarse. En este nivel describimos el flujo normal del proceso, así como queremos que ocurra, sin considerar casos de excepción o errores. El nivel descriptivo nos sirve para validar en forma rápida el alcance del proyecto con los responsables del negocio e introducir al resto de los participantes en él. En el capítulo 3 de este libro se describe como tratar el nivel descriptivo en proyectos de BPMN.

Nivel 2 - Procesos Operacionales: En el nivel operacional se desarrolla toda la lógica de los procesos en su máximo detalle, incluyendo los casos de excepción, fallas e interrupciones que pueden ocurrir a nivel de negocio. La habilidad del analista de procesos consiste en desarrollar un modelo en el nivel 2 que abarque toda la lógica a nivel de negocio y que sea transferible al siguiente nivel de implementación.

Nivel 3a - Modelo Técnico: El modelo técnico es la representación del modelo operacional en un Process Engine, pero adaptando el proceso de negocio a un modelo ejecutable y enriqueciéndolo con aspectos técnicos. Como en realidad no siempre se implementan los modelos de negocio con un Process Engine, dividimos este nivel en un modelo técnico (3a), en el cual en un principio se sigue detallando el modelo de negocio en un BPMS y la opción de un desarrollo propio (3b). En BPMN podemos especificar el modelo técnico directamente en un Process Engine a partir de BPMN 2.0. En el capítulo 5 vamos a explicar como se interpretan los símbolos y patrones de BPMN 2.0 en un modelo técnico.

Nivel 3b - Especificación para Desarrollo: Si no se utiliza un Process Engine, la lógica de negocio tiene que ser desarrollada en algún lenguaje de programación. En estos casos hay que elaborar una especificación técnica, que no tiene mucha relación con BPMN. Los diagramas deben pasarse a una especificación adecuada para el ambiente de programación escogida. En la sección 4.4.5 vamos a tratar en forma abreviada este tema, pero hoy en día ya no tiene mucha importancia debido a la gran oferta que existen el mercado de plataformas de BPMS.

Nivel 4b - Implementación: Luego de la especificación del nivel 3b es necesario implementar técnicamente el proceso en una plataforma «tradicional». Si se utiliza un Process Engine no es necesario elaborar una especificación para el desarrollo, razón por la cual el esquema de la pirámide se muestra en forma asimétrica.

El marco estructural de BPMN presentado, es genérico y no depende de ningún tipo de herramientas de software. El marco nos orienta en la guía práctica de este libro y abarca más de la mitad del volumen de esta obra, describiendo casos de uso, ejemplos y en general buenas prácticas para los diferentes patrones que se dan modelando procesos.

Capítulo 2

La Notación en Detalle

2.1. Entender BPMN

«¿Qué sabe el mono del gusto del Ingwer?»

Este refrán de la India hace alusión a que «si alguien no puede entender algo, tampoco lo puede valorar». Suponemos que el equivalente sería: «tirarle perlas a los chanchos».

Podríamos suponer que también BPMN es una perla que no cualquiera sabe valorar porque no cualquiera lo puede entender. Por esta razón le pedimos tomarse algo de tiempo, para afianzarse con los principios fundamentales de este estándar, no se arrepentirá. Una vez que haya entendido realmente lo que BPMN es, va a disponer de un entorno poderoso que le será de gran beneficio en proyectos de BPM.

Si usted ya conoce la especificación de BPMN, no va a encontrar muchas cosas nuevas en este capítulo. Lo que si encontrará son explicaciones e interpretaciones que no encontrará en la especificación original y que describimos en forma más extensa y con ejemplos las convenciones de la simbología.

Si usted piensa que conoce BPMN bien o si ya cuenta con experiencia práctica en el uso de la notación, le será útil la lectura de este capítulo. La experiencia nos ha mostrado, que muchos aún no han asimilado los principios básicos de la notación, como por ejemplo que los flujos de secuencia no se deben utilizar como conexiones entre los pools.

19

2.1.1. Lo que BPMN debe cumplir y lo que no

BPMN fue desarrollado para modelar procesos. Esta afirmación suena banal, pero en muchas ocasiones se critica que BPMN no puede representar las siguientes estructuras:

Mapas de procesos
Estructuras organizacionales
Estructuras de datos
Estrategias y modelos de negocio
Reglas de negocio

Infraestructura de TI

BPMN se concentra en el modelamiento de los procesos y no de otras estructuras organizacionales. BPMN no fue concebida como una notación para modelar otras estructuras de la arquitectura empresarial. Nos parece bien que así sea; imagínese la complejidad que tendría, fuera de la que tiene, si pretendiese abarcar la metodología y las reglas sintácticas para todos los otros modelos que describen una organización.

También nosotros estamos conscientes que no es suficiente aplicar sólo BPMN para introducir BPM en una organización. Muchos expertos de BPM, sobre todo aquellos que vienen del mundo de ARIS y han utilizado la técnica de «event process chain (EPC)» se quejan que BPMN no es suficiente. Esta falta de entendimiento se debe principalmente a que no comprendieron los objetivos del estándar de BPMN:

Modelos de BPMN pueden relacionarse con otros modelos de una arquitectura empresarial.

BPMN ofrece la posibilidad de ampliarse, por ejemplo de incluir símbolos propios o de relacionarse con otros objetos de una arquitectura empresarial.

Admitimos que sería mejor si es que existiera una convención que fuera más allá de sólo contemplar la vista de los procesos. La integración a una arquitectura empresarial no lo considera el estándar, pero tampoco lo impide. Así algunas empresas han hecho esta integración en sus plataformas de arquitectura empresarial.

2.1.2. Los elementos básicos de BPMN

Cualquier objeto que usted utilice de BPMN en sus diagramas puede relacionarse con las categorías mostradas en la figura 2.1. A estas categorías se les llama en BPMN elementos básicos de la notación.

En un principio en un proceso hay que hacer ciertas cosas (actividades), pero a lo mejor sólo bajo ciertas condiciones (Gateways) y además pueden ocurrir cosas (eventos). A estos objetos se les denomina en BPMN objetos de flujo y se conectan por medio de un flujo de secuencia, pero sólo dentro de un pool, o lanes dentro de un pool. Si se requiere una relación entre dos o más pools se utilizan flujos de mensaje.

Además existen objetos llamados *artefactos* los cuales enriquecen de información la descripción de un proceso, pero los cuales no tienen ninguna influencia en la lógica del proceso. Cada artefacto puede relacionarse con cualquier objeto de flujo a través de objetos del tipo *asociación*. También está permitido utilizar símbolos propios como artefactos.

En BPMN 2.0 se incluyó una nueva categoría de objetos, la categoría datos.

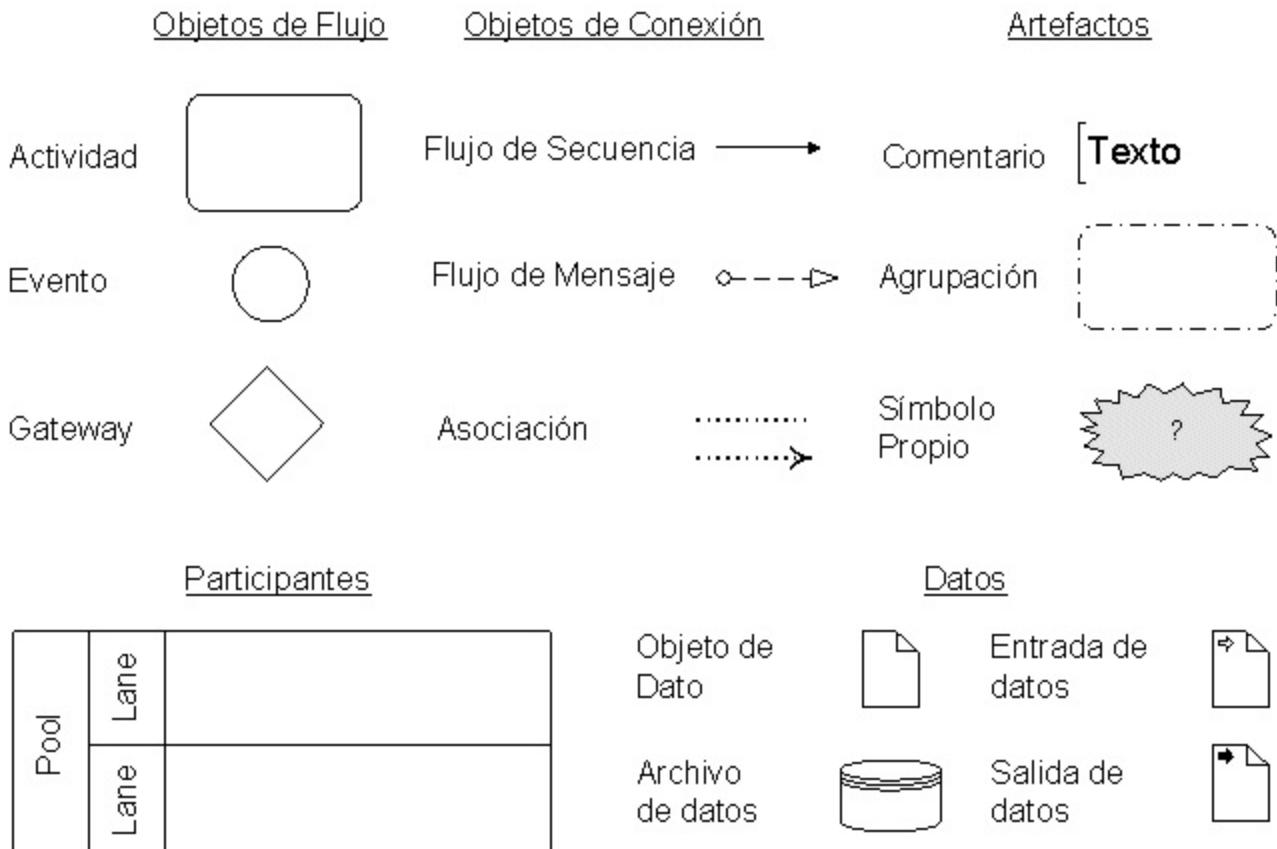


Figura 2.1: Elementos básicos de BPMN

Si en un proceso es importante mostrar como va cambiando de estado (información) un objeto de negocio (solicitud, contrato etc..) podemos utilizar el objeto de datos y relacionarlo con el objeto de tipo asociación a las actividades. Ocasionalmente también se puede asociar a otros tipo de objetos de flujo.

Pues bien, podríamos terminar aquí diciendo que este es el esquema básico de BPMN, pero si somos sinceros faltan algunas consideraciones importantes para entender BPMN:

Las reglas sintácticas que se esconden detrás de este simple esquema

La clasificación de los objetos Responder a las preguntas de cómo utilizo toda esta combinatoria en proyectos reales

Los dos primeros puntos los vamos a tratar en este capítulo y el tercer punto se refiere a conocimientos adquiridos en la práctica (soft-skill, buenas prácticas). Con la finalidad de ahorrarle tener que pasar en parte por esta larga curva de experiencia, por la que pasamos nosotros, incluimos a este libro otros capítulos en donde describimos en forma estructural nuestro conocimiento adquirido en la práctica.

2.1.3. Las diferentes vistas de un proceso

Si alguien ha modelado procesos con otras notaciones, al comienzo le va a ser difícil acostumbrarse a un principio importante de BPMN: «Todo es un aspecto de la

perspectiva».

BPMN parte de la base que en un diagrama pueden representarse a uno o más participantes, pero el lector ha de tener cuidado de no confundir un participante con un rol, un departamento o usuario. Un participante es para BPMN en primer lugar un elemento lógico, cuya aplicación obedece a las siguientes reglas:

En un proceso existe un sólo participante (Este principio confunde a menudo al lector)

Este participante posee el control absoluto sobre la lógica del proceso Otros participantes no pueden influenciar este proceso, en algunas ocasiones ni siquiera saben como está organizado

El participante es por definición el responsable del proceso

Si varios participantes deben interactuar con otros procesos, deben de hacerlo por medio del intercambio de información (flujo de mensaje), información que lógicamente apoya la operación del proceso

Debido a estos principios, se da que cada participante tenga su propia vista sobre el proceso general, es decir diferentes perspectivas. Este hecho nos lleva a deducir que un proceso de negocio puede y por lo general tiene varios modelos de procesos, tantos procesos como participantes existan. El objeto que en BPMN representa un participante es un pool.

Si usted aprende a utilizar bien los pools, habrá entendido uno de los principios más importantes del modelamiento de procesos (con BPMN), por lo menos si persigue el objetivo de introducir BPM en su organización y lograr una integración entre la capa de negocio y la de tecnología. En la sección 2.9 vamos a tratar en forma más extensa este tema.

2.1.4. Modelos, instancias, token y correlaciones

En el documento de especificación BPMN 2.0 de la OMG encontrará en el capítulo 7 «Resumen» una sección que no existía anteriormente en la versión 1.2: «Entender el comportamiento de los diagramas». El documento se refiere a entender el comportamiento de los procesos en los diagramas. Como un diagrama puede contener varios pools, se entiende que un diagrama puede contener n procesos. Esta representación tiene sentido de acuerdo al concepto de BPMN, pero no es siempre fácil de entender. Algunos modelos son tan complejos que el lector pierde el hilo de la lógica representada. Sin embargo, si entiende los siguientes principios y conceptos, le será bastante más fácil de comprender:

Modelo de procesos: En un diagrama pueden representarse uno o más modelos de procesos. Cada modelo constituye la descripción de un proceso.

Instancia de proceso: Bajo instancia de proceso se entiende un proceso concreto en la realidad, es decir casos reales como por ejemplo la reclamación de un cliente crea una instancia del proceso de reclamaciones. Algunos procesos se instancian sólo un par de veces al año y otros más a menudo, por ejemplo la solicitud de información sobre el registro de infracciones, se instancia 90 millones de veces en el año.

Token (marca): El concepto de marcas se utiliza para visualizar y probar el comportamiento de los procesos diseñados. Las marcas recorren en forma de una animación la lógica por los flujos normales y los de excepción. El token se puede definir como una construcción teórica que podría compararse con un mapa carretero o de una ciudad sobre el cual uno pone un auto de juguete y desplaza este desde un lugar a un destino. Si el auto llega a un cruce el chofer tiene que tomar la decisión que camino seguir dependiendo de la ruta, el tráfico etc.. Es posible que el auto llegue a un cruce donde debe seguir derecho y además doblar porque tiene que dejar pasajeros en dos puntos diferentes. En este caso debería ser clonado, lo que lógicamente no es real. En la realidad los pasajeros van a tomar otro vehículo mientras uno sigue a su destino, o bien el vehículo va primero a un lugar, se vuelve y luego va al otro. En todo caso en nuestro ejemplo el mapa representa el modelo de proceso y una ruta representa un caso de recorrido que puede tomar una instancia. Si usted aplica el concepto de token va a poder entender cualquier modelo de proceso, incluso los más complejos. El concepto de token se describe también en la nueva especificación de BPMN. En varias ocasiones vamos a utilizar este concepto en nuestros ejemplos para una mayor comprensión.

Correlación: Seguramente usted ha recibido muchas veces correspondencia de instituciones que contienen un número de referencia, número de acta, número de ticket, etc.. Si usted responde a la institución tiene que indicar esta referencia para que pueda ser identificada. A esta asignación en forma de un identificador se le llama correlación. La correlación es importante para los procesos tanto desde el punto de vista organizacional como técnico. Un caso siempre debe estar bien correlacionado para que no ocurran errores o pérdidas de datos. Desgraciadamente en muchas ocasiones por negligencia en temas de correlación se han cometido errores que han sido caro de subsanar.

2.1.5. BPMN en español

En las primeras capacitaciones que sostuvimos de BPMN en Alemania, se utilizó sólo la nomenclatura de BPMN en inglés. Los usuarios de negocio que estaban cambiándose de la notación EPC a BPMN, estaban intranquilos y uno de los participantes interrumpió al relator diciéndole:

¿De que diablos habla usted? ¿Error? ¿Catchen? ¿Throwen? ¡Esa terminología la

conozco de nuestros programadores, ellos hablan siempre así! ¡Pero nosotros no somos programadores, somos gente de negocio! ¡Esta idiotés no la necesitamos, no es para nosotros!

La audiencia se sintió interpretada y apoyó la posición de este atrevido usuario de negocio. En esta ocasión nos costó recuperar el control en la clase y explicarles que esta simbología también estaba pensada como instrumento de diseño y comunicación para la capa de negocio. En la siguiente capacitación, experimentamos el grado de aceptación traduciendo en gran parte la terminología y el resultado fue positivo, incluso se identificaban con los escenarios modelados.

Debido a esta experiencia nos reunimos con círculos profesionales y universitarios que se dedican al tema y acordamos un glosario alemán para la simbología de BPMN.

En la región de habla española estamos más acostumbrados a una mezcla idiomática anglosajona, porque gran parte del conocimiento se traduce del inglés y muy pocas veces de otros idiomas, pero eso no significa si se exagera en el uso de términos en inglés que, puede igualmente haber problemas de aceptación. Eso 2.2.

ACTIVIDADES SIMPLES Y EVENTOS INDEFINIDOS 25

si, en nuestra lengua tenemos el problema que según la región (España, México, Sudamerica), no es tan fácil de llegar a un acuerdo general porque contamos con varias filosofías de como usarlas. En España existe una tendencia hacia una traducción total, mientras que en México (más cerca de EEUU) se tiende a dejar muchas palabras en su expresión original. Sólo en la pronunciación de la palabra se españoliza. Nosotros hemos optado en el glosario por una mezcla sana, dejando el nombre original de los objetos básicos del BPMN y traduciendo aquellos que no son tan comunes o más técnicos. El glosario que proponemos en este libro lo encuentra en el capítulo 8.

2.1.6. Simbología y atributos

La especificación de BPMN no sólo define los símbolos que se utilizan en el modelamiento de procesos, sino que además existen una serie de atributos que se pueden relacionar o asignar a los objetos. Muchos de estos atributos no se muestran o visualizan en los diagramas y se encuentran en los atributos de descripción de los objetos. La razón de porqué no se muestran siempre o sólo a pedido si la herramienta de modelado lo permite, es porque estos atributos apoyan la implementación técnica de los procesos.

2.2. Actividades simples y Eventos indefinidos

La figura 2.2 muestra un proceso muy simple. El proceso es gatillado cuando alguien manifiesta su hambre, por lo que gatilla la actividad de comprar alimentos y en secuencia lógica le sigue la actividad de preparar una cena. Al final la cena se lleva a cabo y los participantes se manifiestan satisfechos.

En este diagrama podemos identificar fácilmente los siguientes objetos y su semántica:

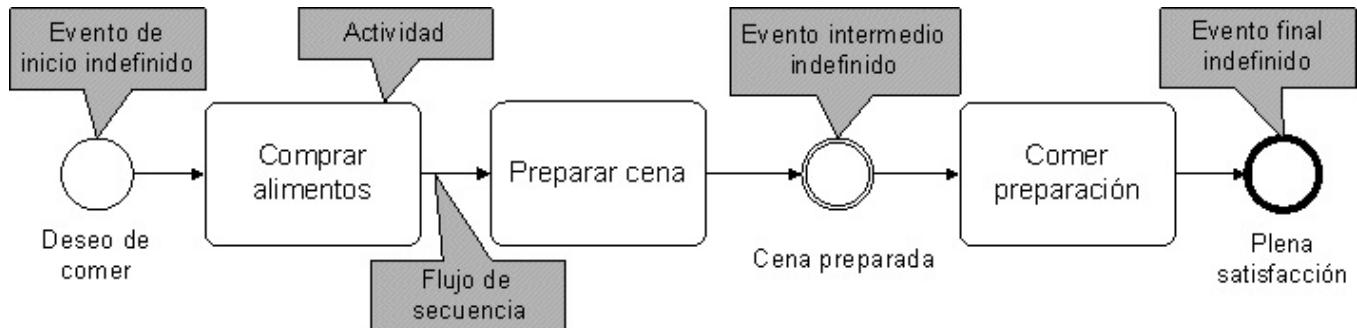


Figura 2.2: Nuestro primer proceso

Actividades: Las actividades son la espina dorsal de los procesos, debido a que son las actividades las que transforman el estado de un objeto de negocio para que el proceso pueda llegar a producir valor para los clientes.

Nuestro Tip en BPMN

Las actividades se pueden definir como «acción sobre un objeto», es decir una actividad se denomina siempre con un verbo (acción) y un sustantivo (objeto). Por ejemplo «comprar alimentos» y no «primero hay que comprar los alimentos».

Eventos: Eventos indican que al inicio, en forma intermedia o al final del proceso algo significativo ocurrió. Por el momento hablaremos sólo de eventos indefinidos, más adelante vamos a conocer otros tipos de eventos.

Eventos de inicio nos indican que tipo de ocurrencias suceden para que un proceso comience.

Eventos intermedios muestran un estado que el proceso ha alcanzado y que en el modelo por alguna razón lo queremos retener. No se utilizan muy a menudo, pero pueden ser muy útiles, por ejemplo si el estado representa un hito y se quiere medir el tiempo transcurrido hasta alcanzar el hito.

Eventos finales indican que se logró al finalizar una trayectoria del proceso. Junto a esta clasificación fundamental tenemos otros aspectos que diferenciar en los eventos:

Eventos de inicio son eventos de captura (en inglés: catching events), es decir algo independiente del proceso ocurrió, pero el proceso tiene que reaccionar o esperar.

Eventos intermedios pueden ser del tipo de captura o pueden ser impulsados por

alguna actividad del mismo proceso (en inglés: throwing events). El evento intermedio indefinido representa un estado intermedio que ha alcanzado un proceso, por consiguiente se trata de un evento del tipo impulsado. Más adelante vamos a conocer otros tipos de eventos intermedios, los cuales vamos clasificar como del tipo de captura.

Eventos finales ocurren de forma que el proceso ya no puede reaccionar a ellos, por lo tanto los clasificamos como del tipo impulsados.

Nuestro Tip en BPMN

Eventos representan ocurrencias, es decir algo ocurrió (tiempo pasado) en forma independiente del proceso o bien eventos impulsados por alguna actividad del proceso. Debido a esta definición denominamos los eventos con el objeto y ocupamos el verbo en participio, por ejemplo «orden recibida». La especificación de BPMN no obliga usar un evento de inicio y de fin, usted los puede omitir. Pero si decide utilizar un evento de inicio está obligado a usar uno de término y vice versa. Nosotros utilizamos siempre eventos de inicio y de fin, porque de esta forma podemos retener la causa del inicio y podemos indicar las diferentes trayectorias como termina. Sólo en los subprocessos desistimos de esta posibilidad, pero más adelante explicaremos por qué.

Flujo de Secuencia: El flujo de secuencia describe la secuencia temporal y lógica en el cual se combinan los elementos de flujo, es decir las actividades, eventos y Gateways (los cuales explicaremos en seguida). El flujo de secuencia es también la trayectoria del proceso por el cual marcha el token. El token «nace» junto a una instancia con el evento de inicio. A través del flujo de secuencia se llega a las actividades, a los estados intermedios y al evento final, donde es «consumido» y desaparece, al mismo tiempo «muere» nuestra instancia.

Nuestro Tip en BPMN

Si usted quiere, puede diagramar los modelos de procesos en forma vertical (hacia abajo), en vez de como en todos nuestros ejemplos en forma horizontal. No es normal diagramar la línea de tiempo hacia abajo, pero no está prohibido. Nosotros indicamos la línea de tiempo, como la mayoría, de izquierda a derecha.

2.3. Flujo de Procesos con Gateways 2.3.1. Gateway exclusivo de datos (XOR)

Casi ningún proceso tiene un flujo uniforme. En la mayoría de los casos, las instancias recorren diferentes trayectorias, dependiendo de las condiciones y reglas que apliquen.

En nuestro ejemplo simple (figura 2.3) queremos entretenernos con el arte de cocinar. Impulsados por el hambre, pensamos que queremos comer hoy. Debido a

que sólo conocemos tres recetas, nos decidimos por una de ellas, que puede ser cocinar pasta, freír un steak (trozo de carne frita) o preparar una ensalada. Estas tres posibilidades son excluyentes, es decir sólo una opción es válida. Usted pensará ¿pero y si queremos un plato de fondo y la ensalada también? ¡Tenga paciencia ya hablaremos al respecto!.

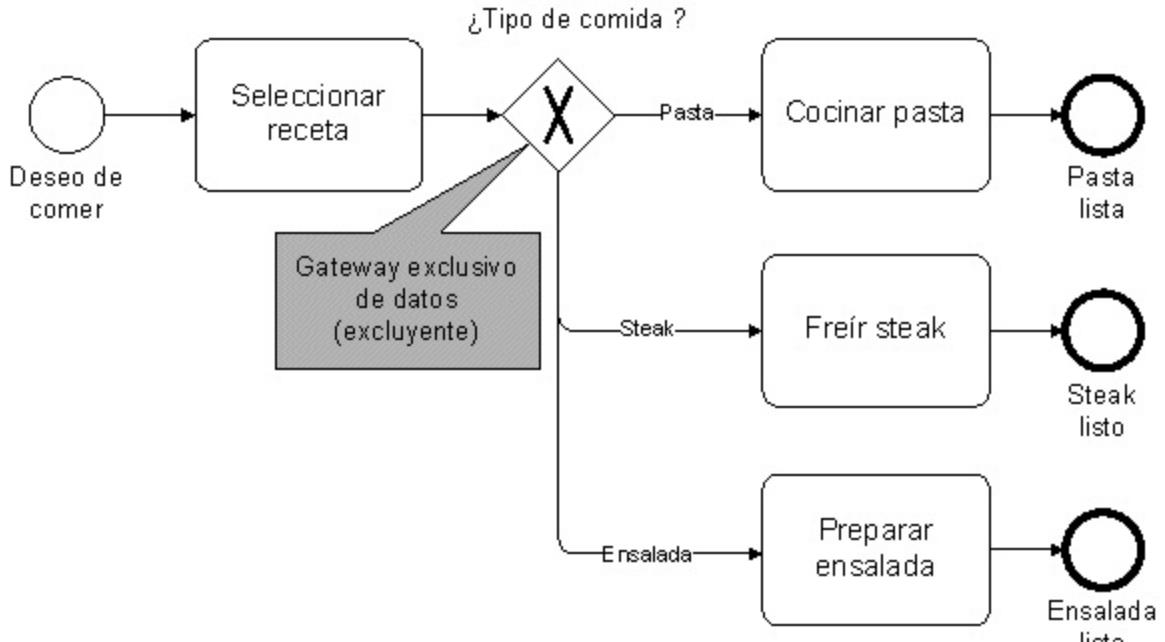


Figura 2.3:

XOR-Gateway o gateway exclusivo de datos

La compuerta en que tenemos que tomar la decisión en BPMN se denomina Gateway. Como esta decisión la tomamos de acuerdo a la información recibida (la receta escogida) y la compuerta permite recorrer sólo una alternativa, hablamos de un «Gateway exclusivo de datos», también en forma abreviada de un «Gateway XOR» (XOR = Exclusive OR).

¡Cuidado, no confunda un Gateway con una Actividad! El Gateway requiere de un hecho (una variable). La actividad es la encargada de producir este hecho y disponibilizar la variable para el Gateway. Esto parece ser un principio diferenciador común, pero a usted no se le olvide cuando esté modelando. Vamos a retomar este tema, cuando hablemos de los Business Rules Management Systems o sistemas de reglas de negocio (ver capítulo 4).

Nuestro Tip en BPMN

Sobre el Gateway posicionamos la pregunta sobre la cual tiene que tomarse la decisión. Los diferentes respuestas que pueden darse la posicionamos en los flujos de salida que se dan. La especificación indica que sólo los flujos deben nombrarse. La pregunta aclaratoria es parte de nuestra recomendación. La cantidad de flujos de salida es según la especificación ilimitada y no importa donde se posic和平n estos flujos, en el vértice o arista del Gateway.

En BPMN existen para el XOR-Gateway dos símbolos que tienen el mismo

significado (ver figura 2.4). En este libro ocupamos el símbolo que contiene el «X» porque lo consideramos más expresivo, pero usted puede escoger el que mejor le parezca o el que apoye su herramienta. De todas formas no recomendamos mezclar ambos símbolos en los diagramas; decídase a utilizar uno de ellos por convención.

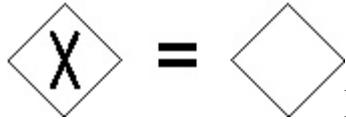


Figura 2.4: Símbolos equivalentes a XOR-Gateway

No es inusual que un proceso tenga varios eventos de término, como en la figura 2.3. En este caso el proceso puede producir tres estados diferentes de término. En diagramas complejos ayuda bastante a comprender que un proceso puede terminar con diferentes estados. Más adelante vamos a conocer otras razones de porque tiene mucho sentido emplear diferentes eventos de término. Si los diferentes recorridos tienen su termino individual o si hay que unirlos a un término común depende de la lógica del proceso. BPMN permite ambas situaciones. Si queremos unir los recorridos porque le sigue una actividad que debe emplearse para todos los estados anteriores también podemos usar el XOR-Gateway como unión técnica. La unión técnica asegura que el token que viene de cualquiera de los recorridos, sea guiado hacia un flujo común (ver figura 2.5).

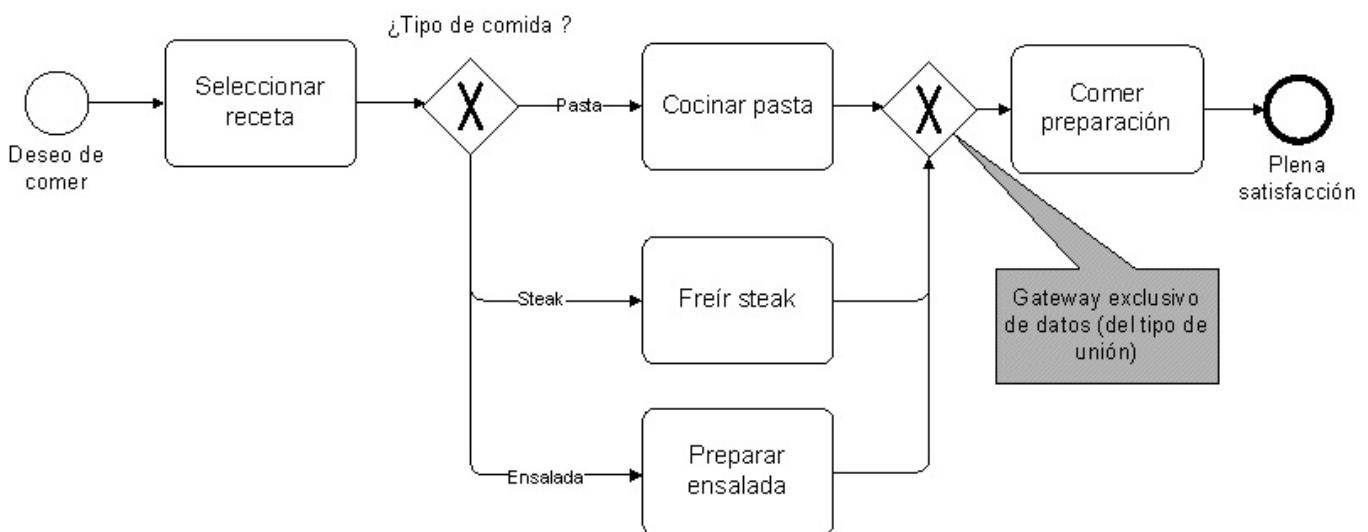


Figura 2.5: Modos de utilización de XOR-Gateway

La posibilidad que tenemos de utilizar el XOR-Gateway como elemento de bifurcación (XOR-Split) o unión (XOR-Join) puede que al principiante lo confunda. La notación permite incluso usar el XOR-Gateway como una unión de entrada y bifurcador de salida con un sólo símbolo (vea figura 2.6).

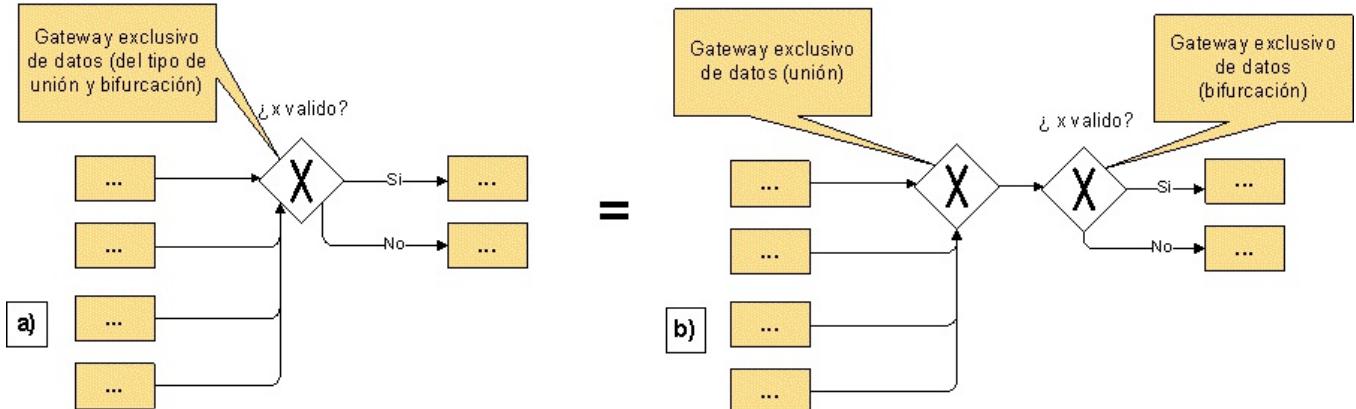


Figura 2.6: Utilización de XOR-Gateway como a) XOR-Split y XOR-Join de forma compacta y b) XOR-Split y XOR-Join de forma separada

Si quiere utilizar esta posibilidad de modelamiento para compactar sus diagramas, tiene que decidirlo usted. Nosotros desistimos de esta posibilidad y usamos mejor dos XOR-Gateways seguidos para evitar interpretaciones ambivalentes.

2.3.2. Gateway paralelo

¿Cómo podemos modelar el caso que queremos la ensalada como acompañamiento? Representemos el caso más sencillo que queremos, el acompañamiento de todas formas, entonces podemos modelar el proceso como muestra la figura 2.7.

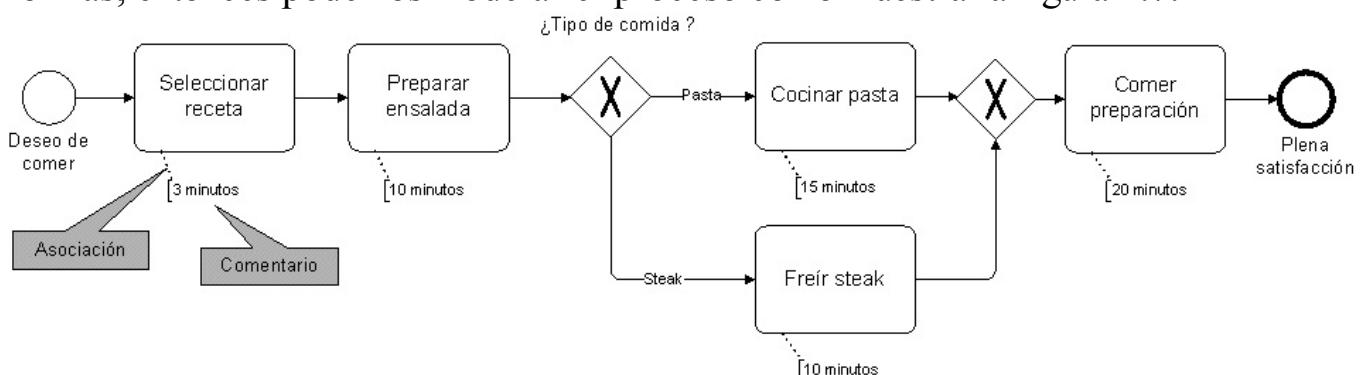


Figura 2.7: Preparación de ensalada y plato de fondo

En esta ocasión aprovechamos de incluir un nuevo objeto, el comentario de texto. Este objeto pertenece a la categoría de los «artefactos», el que podemos asociar a cualquier objeto de flujo (en este caso a una actividad). En este ejemplo indicamos el tiempo promedio que demora cada actividad. La suma de los tiempos de cada una actividad nos arroja el tiempo de ciclo del proceso completo, en este caso 48 minutos, si nos decidimos por cocinar pasta y 43 minutos si nos decidimos por los steaks. ¡Felicitaciones!, acaba de realizar su primer análisis de indicadores de tiempo de ciclo.

Según este proceso, requerimos entonces de 23 a 28 minutos, hasta que podamos sentarnos a comer. Si tuvieramos mucha hambre, podría volverse un calvario tener que esperar tanto para comer. ¿Qué podemos hacer? Lógicamente un rediseño del

proceso. Podríamos pensar en paralelizar la preparación de la comida y de esta forma acelarar el proceso en general, es decir preparar la ensalada al mismo tiempo que el plato de fondo. El objeto que vamos a usar para este caso se denomina «AND-Gateway» el cual apreciamos en la figura 2.8.

El paralelismo no significa que obligatoriamente las actividades tienen que ejecutarse exactamente en forma simultánea, pero pueden comenzar cuando la condición se de. El diseño de la figura 2.7 obliga a preparar la ensalada antes de comenzar a cocinar, lo que no necesariamente tiene que ser así, por ejemplo mientras la pasta está en la olla, podemos preparar la ensalada. De acuerdo a nuestro nuevo cálculo podríamos acortar el tiempo de ciclo en 10 minutos. Como usted se puede imaginar, este caso es clásico en la optimización de procesos, en el sentido de intentar de paralelizar al máximo posible.

En el ejemplo descrito el proceso no sólo se paralleliza (AND-Split), sino que más adelante se sincronizan los dos flujos (AND-Join). La razón debería quedar clara: sólo al estar lista todas las preparaciones, podemos sentarnos a comer, de lo contrario y según el proceso iríamos comiendo cada preparación en cuanto esté

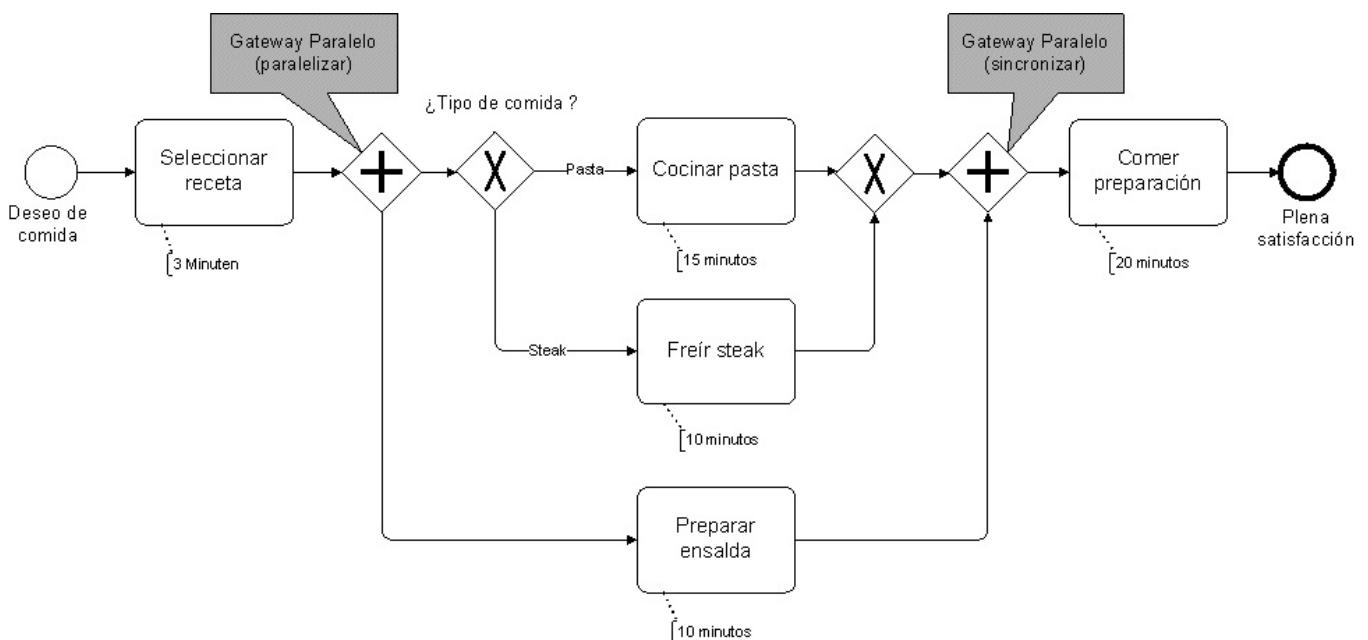


Figura 2.8: Preparación de ensalada junto con plato de fondo

lista, lo que no tiene mucho sentido.

¿Cómo se comportaría en una instancia de este proceso el concepto de token?

El token nace con el evento de inicio, luego recorre la actividad seleccionar receta y llega al Gateway paralelo (AND-Split). En un Gateway paralelo se dividen (nacen) tantos token como salidas tiene la compuerta, en este caso son dos. El primer token se dirige al XOR-Gateway, en donde según la receta escogida la compuerta lo dirige a la actividad correspondiente; en este caso asumiremos que será cocinar pasta en donde permanecerá 15 minutos (el tiempo de cocción). Al mismo tiempo el segundo

token se dirige a la actividad preparar ensalada donde permanece 10 minutos, es decir el tiempo que nos demoramos en preparar la ensalada. Pasado estos 10 minutos se dirige al próximo AND-Join en donde espera 5 minutos hasta que llegue el token de cocinar pasta. Es decir la instancia espera en el AND-Join hasta que lleguen ambos token y los fusiona en uno sólo. Enseguida el token fusionado se dirige a la siguiente actividad.

¿Le parece esta descripción del token muy técnico o abstracto? En realidad no lo es, porque se comporta igual que en la realidad: La preparación de la ensalada se demora menos, entonces esperamos hasta que el proceso de cocinar la pasta haya concluido; recién entonces nos sentaremos a la mesa a comer todo lo preparado. ¿Por qué se necesita entonces el concepto del token? Piense entonces en el caso de los 90 millones de transacciones que se realizan al año en el registro de infracciones. Estas consultas no se realizan en forma secuencial, sino en forma concurrente, cada instancia con diferentes flujos. ¿Cómo podríamos reconstruir en caso de error las transacciones fallidas sin un concepto de token? ¿Se habrá dado cuenta que instancias y tokens no son lo mismo? En una instancia pueden haber varios tokens activos y fuera de registrar todo lo que sucede nos indican dónde y en qué estado se encuentra la instancia.

En las siguientes dos preguntas de control queremos probar su grado de entendimiento con respecto al concepto de tokens.

Pregunta 1: En la figura 2.9 le mostramos el mismo proceso descrito anteriormente, pero hemos eliminado el AND-Join y el flujo de la actividad preparar ensalada se dirige ahora directamente al XOR-Join. ¿Cómo reacciona el proceso si creamos una instancia (nos decidimos por cocinar pasta)?

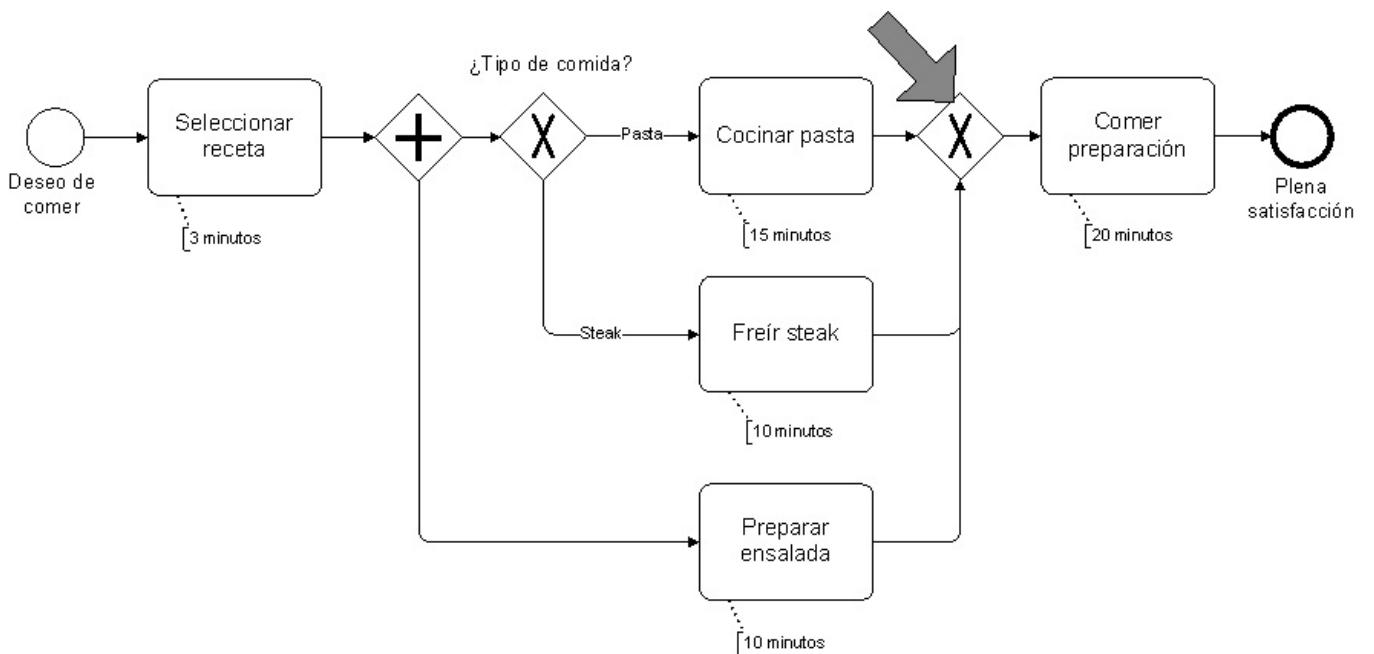


Figura 2.9: Caso And-Split sin And-Join

Respuesta: Con la instancia del proceso se crea simultáneamente el token, el cual es clonado como en el caso anterior en el AND-Split. Apenas esté lista la ensalada se

dirige el token a la actividad comer preparación y ejecuta la tarea, es decir la ensalada se consume sin esperar la pasta. Lo mismo ocurre cuando más tarde la pasta está lista, es decir la actividad comer preparación se ejecuta dos veces, no es precisamente el comportamiento que esperábamos.

Pregunta 2: En la figura 2.10 verá un proceso con sólo dos actividades. El proceso es instanciado. ¿Cuanto tiempo «vive» la instancia?

Respuesta: La instancia «vive» 45 días, que corresponde con el tiempo de ciclo del proceso. A pesar que el primer token se consume luego de 30 días, el segundo token aún está activo durante 15 días más en la actividad 2.

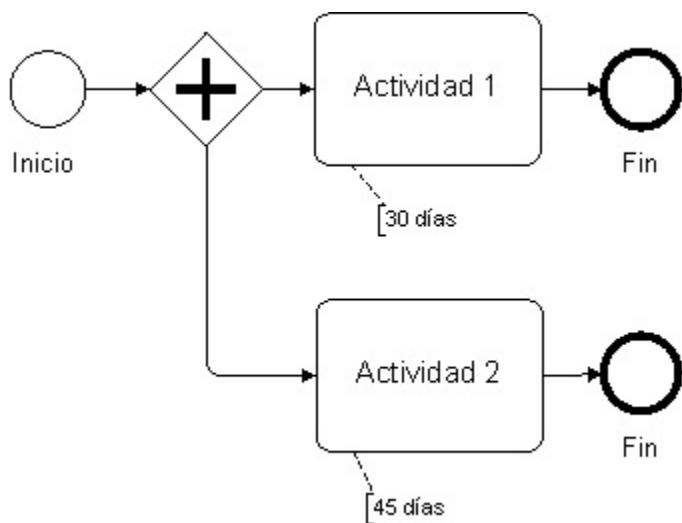


Figura 2.10: Revisión de tiempo de duración de instancias con AND-Split

Recuerde: Mientras exista en un proceso un token activo, vive también la instancia del proceso. Recién cuando «todos» los tokens se hayan consumido, termina o muere la instancia.

2.3.3. Gateway inclusivo de datos (OR)

Para explicar este Gateway más complejo, vamos a ampliar nuestro ejemplo y suponer que queremos diseñar nuestro proceso menos rígido o dicho de otra forma más flexible. Si se da el deseo de comer, queremos modelar la siguiente situación:

sólo preparar una ensalada o,
una ensalada y un plato de fondo, que podría ser pasta o steak o
alguna otra combinación

Con los tipos de objetos que usted conoció anteriormente, podría modelarse como lo muestra la figura 2.11. Si queremos modelarlo en forma más compacta, podemos utilizar el Gateway inclusivo de datos, también llamado OR-Gateway (vea figura 2.12). Con un OR-Gateway podemos formular una situación que responde a las

preguntas «y/o», en la cual podemos escoger uno, muchos o simplemente todos los flujos de salida. Es decir esta compuerta la podemos utilizar para reducir la complejidad y reemplazar preguntas sofisticadas en la combinación de los XORGateway y el AND-Split Gateway. El OR-Gateway también lo utilizamos como unión técnica de todas las combinaciones que se pueden dar con las instancias.

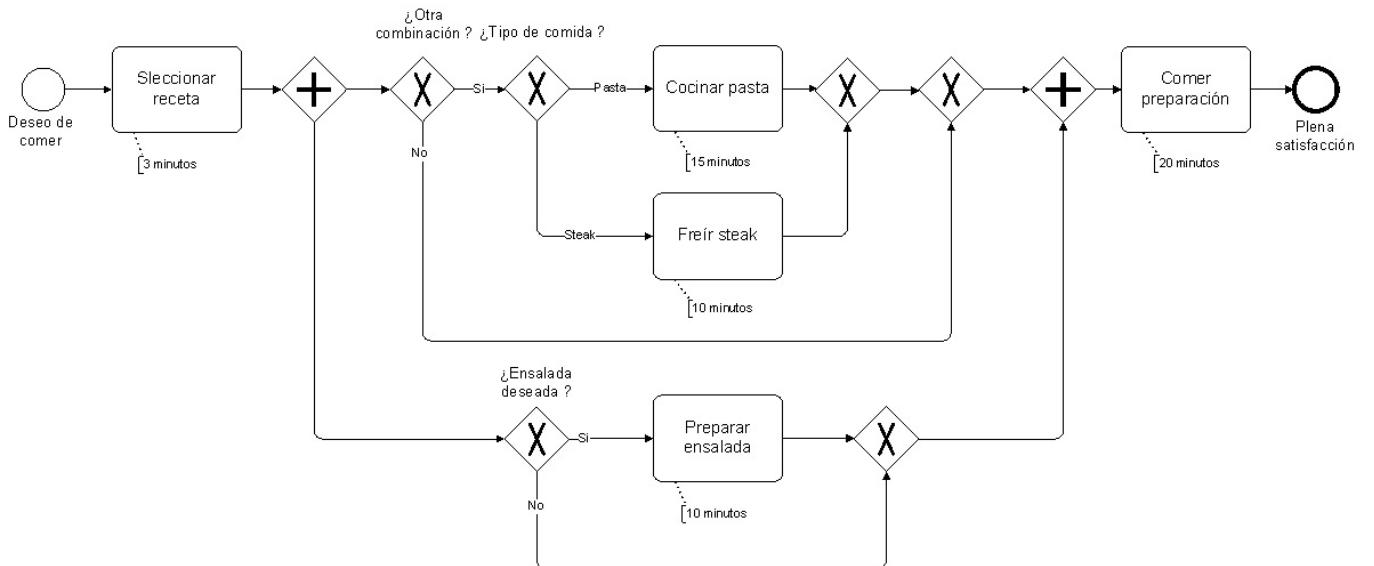


Figura 2.11: Diferentes opciones en la selección de nuestra cena

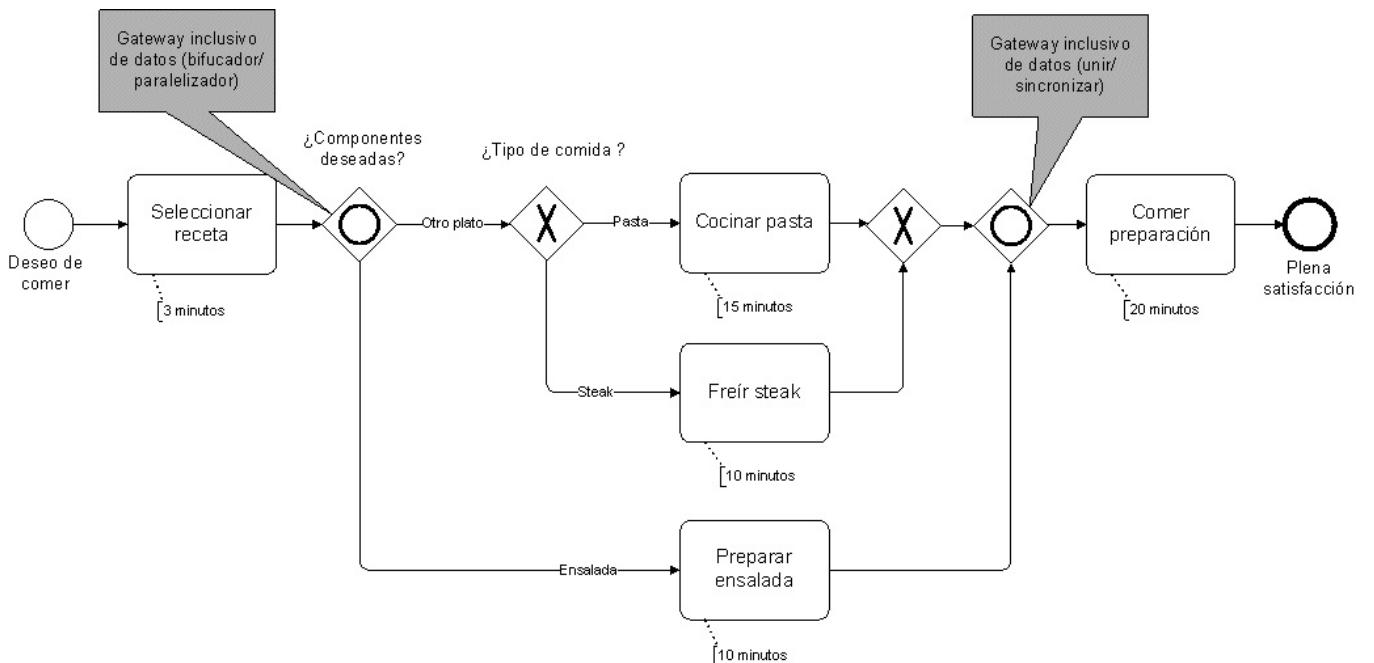


Figura 2.12: Utilización de OR-Gateway como OR-Split y OR-Join

En ambos modelos existe si una diferencia semántica. En la figura 2.11 tenemos la posibilidad de no preparar la ensalada, como también de no escoger ningún plato de fondo, posibilidad que no se da en el modelo de la figura 2.12, pero en el primer caso este posible flujo lleva también al absurdo porque tiene que pasar por la última actividad que obliga a comer lo preparado, a pesar que se desistió de preparar algo para comer. En el caso del OR-Gateway este caso sin sentido no se da, porque la

lógica obliga a decidirse por alguna de las alternativas propuestas. Una simulación por medio de Token, arrojaría en el primer caso un error porque la última actividad no puede ser ejecutada.

Este es un caso sencillo, pero en lógicas complejas ha de tenerse cuidado con el uso del OR-Gateway, sobre todo con las reglas de sincronización, porque pueden darse constelaciones en que no se puede testear fácilmente toda la casuística. Analicemos como ejemplo el caso en la figura 2.13. Dependiendo si después del OR-Split se recorren uno o más flujos, el OR-Join debe sincronizar o no.

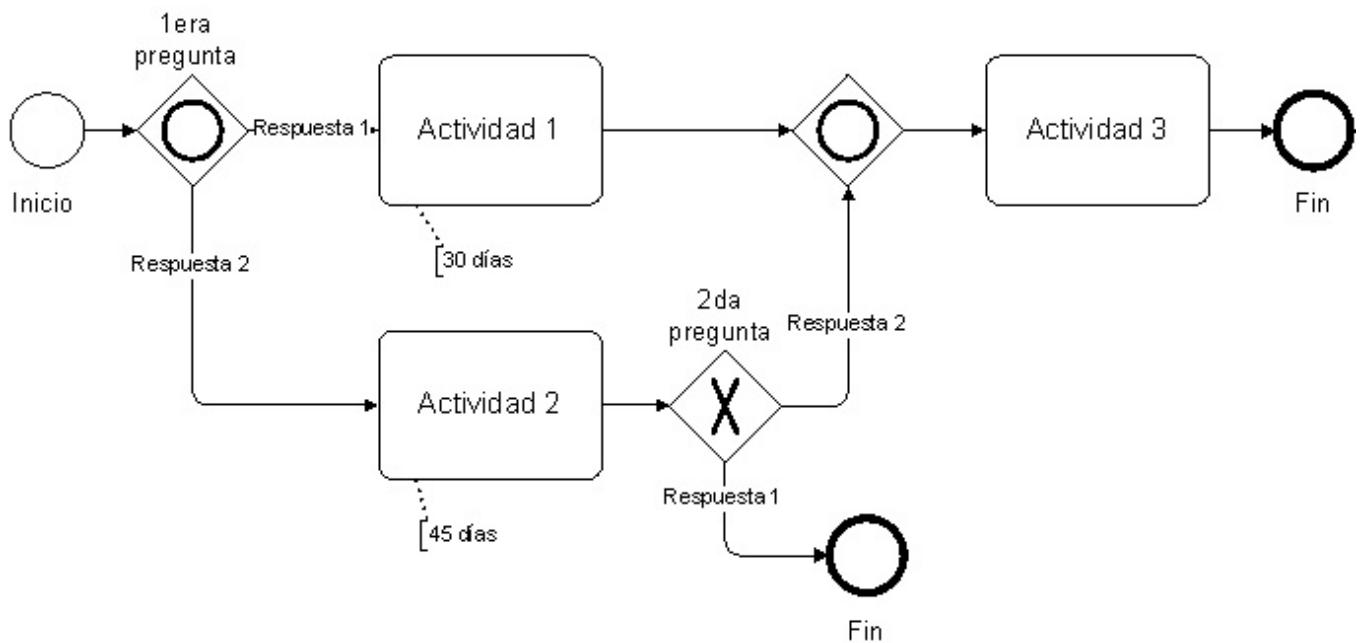


Figura 2.13: Revisión de tiempo de duración de instancias con OR-Gateway en el OR-Join

En la figura 2.13 llega el primer token a los 30 días al OR-Join. Suponiendo que también fue válida la respuesta 2 en el OR-Split, se generó un segundo token que aún permanece 15 días en la actividad 2. Ahora puede suceder que terminada la actividad 2 la decisión en el XOR-Split sea la respuesta 1 y el segundo token sea consumido por el evento de fin. ¿Qué pasa ahora con el primer Token que se queda esperando la sincronización en el OR-Join? El OR-Gateway tiene que registrar, que el segundo token fue consumido y liberar el primero para que pueda continuar el proceso. Si se da un caso como este, la especificación no sabe como resolverlo y arroja un error tipo «exception». Al lector no podemos más que aconsejarle de usar con mucho cuidado el OR-Gateway y si lo hace debe comprobar por medio del concepto de Token, que la casuística no lleve en algunos casos al absurdo.

Sigamos con nuestro caso de cocina y tratemos de modelar todas las opciones de las que hablabamos anteriormente. Si usamos el OR-Gateway tenemos muchas opciones de combinación como lo muestra la figura 2.14, pero también se dan algunas casos absurdos o que no queremos representar.

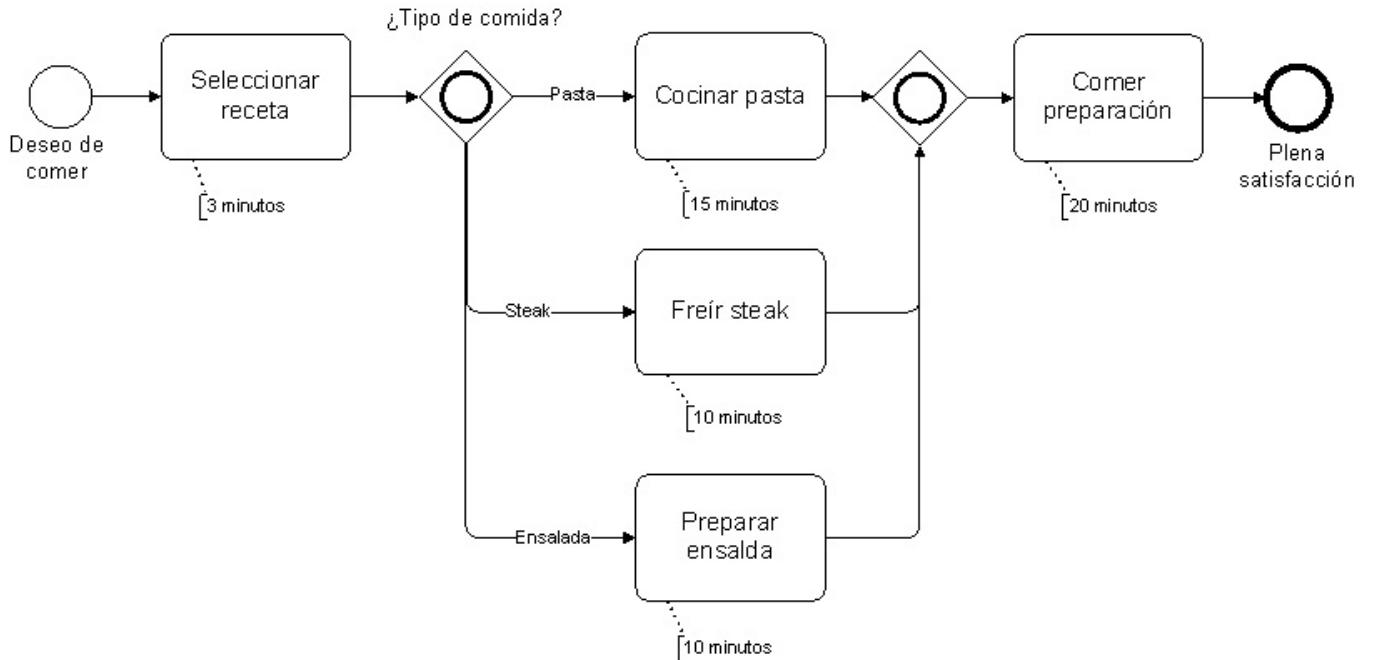


Figura 2.14: Una versión compacta con muchas opciones

Veamos las opciones:

Comemos sólo pasta.

Comemos sólo steaks.

Comemos sólo ensalada.

Comemos pasta y ensalada

Comemos steak y ensalada.

Comemos pasta y steaks.

Comemos pasta, steaks y ensalada.

Las últimas dos alternativas no corresponden con lo que queremos representar o modelar.

2.3.4. El flujo por defecto

Vamos a conocer ahora un tipo de flujo que nos será de mucha utilidad para evitar situaciones contradictorias en el empleo de Gateways del tipo XOR-Gateway u OR-Gateway como los analizados en los casos anteriores. Con el objetivo de mantener nuestros ejemplos sencillos y concentrarnos en el aspecto pedagógico, vamos a desistir por el momento de la opción ensalada.

¿Qué podemos hacer, si no tenemos deseos de comer pasta ni steak? En nuestros modelos anteriores, esta situación nos habría llevado al problema, que nuestro token se habría quedado atascado en el XOR-Split (pregunta: tipo de comida). Para todos estos casos la especificación determina que la herramienta que lo implementa debe arrojar un error del tipo «exception».

Por favor aún no se impaciente, ni se enoje con nosotros, vamos a tratar los casos de excepción (exception handling) más adelante y vamos a explicar que este tipo de objetos de control, no son sólo técnicos.

El tipo de flujo por defecto nos protege ante situaciones indeseadas, como con las que nos confrontamos anteriormente. En la figura 2.15 lo utilizamos, lo reconocerá por una línea en diagonal puesta sobre el flujo. La semántica del flujo por defecto es sencilla: Primero se contemplan todos los flujos salientes con sentido de negocio; sólo si ninguna de la opciones anteriores es válida se emplea el flujo por defecto.

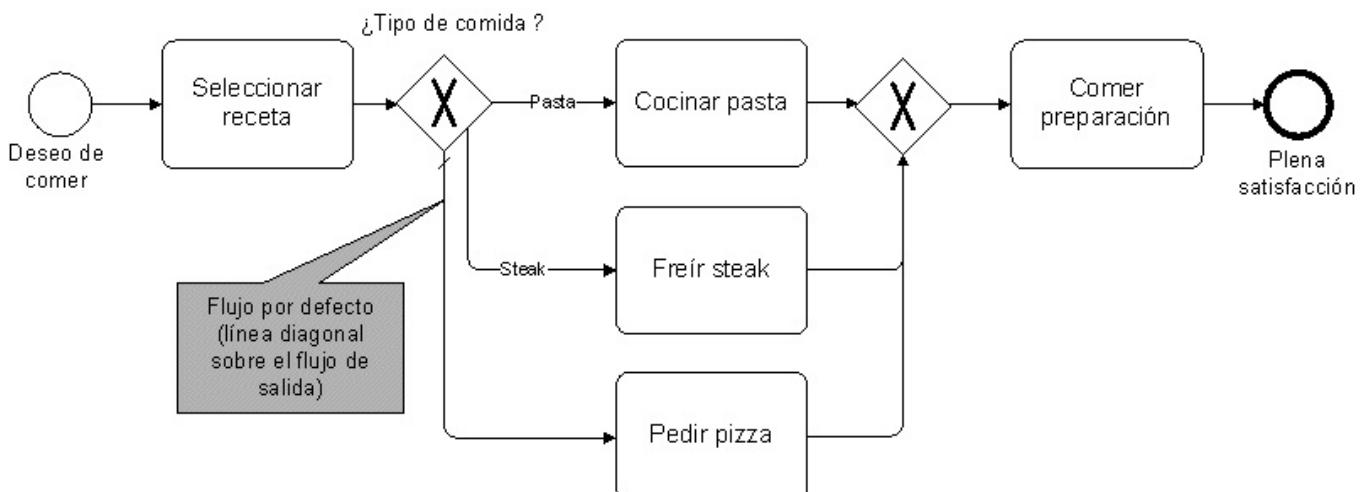


Figura 2.15: Flujo por defecto

2.3.5. Gateway complejo

El Gateway complejo es una compuerta que se utiliza cuando un caso de negocio no se puede representar con ningún otro Gateway. No se utiliza muy a menudo aún cuando existen escenarios donde se podría aplicar bastante bien. Continuando con nuestro ejemplo vamos a suponer que queremos pedir una pizza a domicilio y además de ver el menú de nuestro proveedor de pizza habitual vamos a estudiar algunos otros proveedores que vamos a buscar en internet. Vamos a suponer que tenemos preseleccionada una pizza preferida que habitualmente pedimos a nuestro proveedor, pero vamos a ver si algún otro proveedor nos tienta en probar una pizza nueva. Luego nos decidimos por alguna de las alternativas y hacemos el pedido.

¿Cómo podemos modelar este caso? Si modelamos el caso como muestra la figura 2.16, podríamos sólo pedir la pizza cuando ambas actividades hayan entregado su resultado, lo que no corresponde exactamente con el caso que queremos representar, porque al decidirnos por un nuevo proveedor, queremos hacer el pedido de inmediato. El modelo de la figura 2.17 tampoco es una alternativa, porque de acuerdo al concepto de token, haríamos un pedido siempre dos veces. Si tratamos de modelar este caso con un OR-Join como muestra la figura 2.18, tampoco muestra el comportamiento deseado. En este caso el OR-Join se comporta como AND-Join.

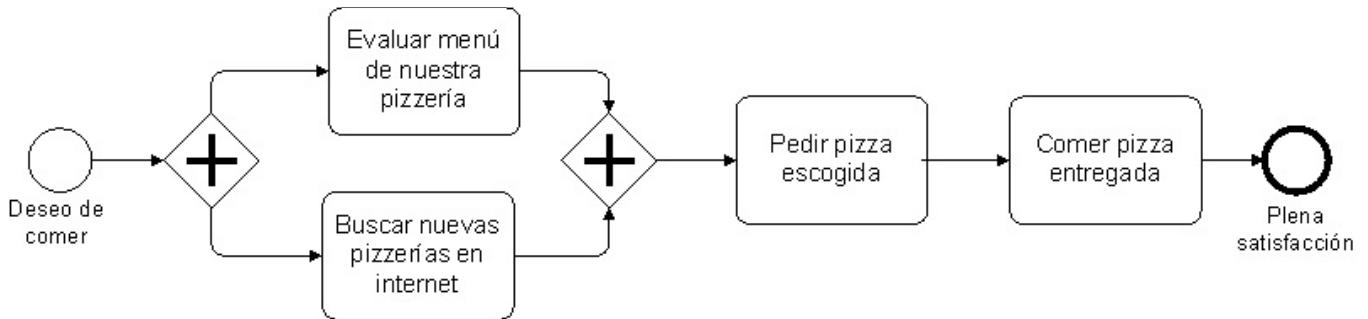


Figura 2.16: Pedido de pizza con un AND-Join

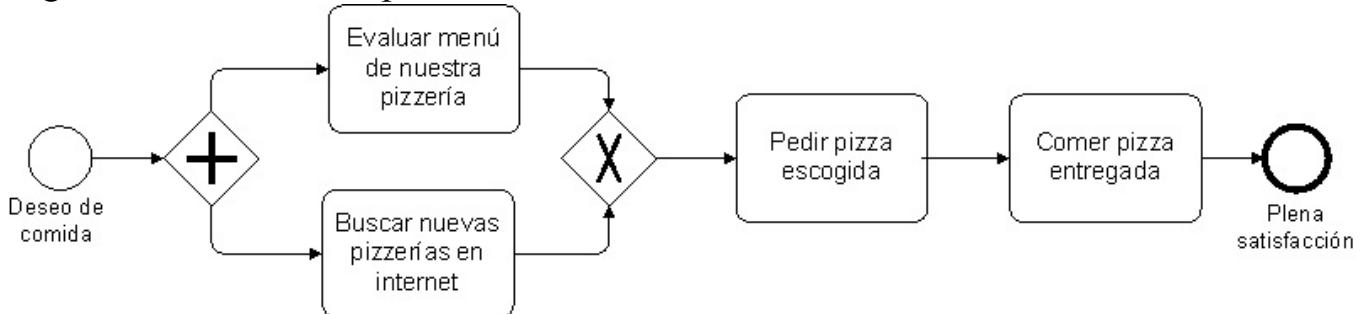


Figura 2.17: Pedido de pizza con un XOR-Join

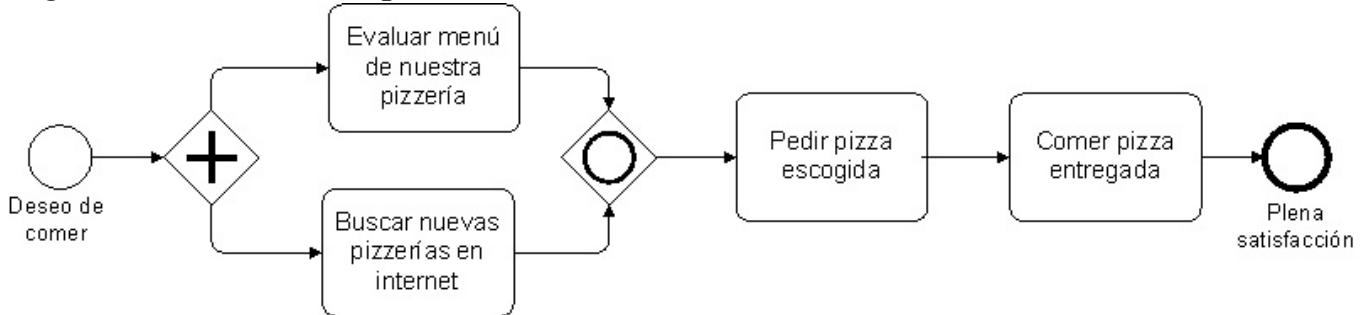


Figura 2.18: Pedido de pizza con un OR-Join

La solución la encontraremos utilizando un Gateway complejo (Complex-Gateway) en combinación con un comentario como muestra la figura 2.19. En el momento de tomar la decisión, por cualquiera de ambas alternativas, pedimos la pizza y desecharmos la otra opción. Esta condición no la puede evaluar la lógica del concepto de token por si sola, por lo que esta condición tiene que ser incluida adicionalmente en un modelo técnico. El comportamiento del token en nuestro caso con un Gateway complejo del tipo join es: En el momento que la compuerta Complex-Join reciba la primera variable de decisión de alguna de las actividades, el token se dirige a la actividad «Pedir pizza» y consume (hace desaparecer) el próximo token activo. Este es entonces exactamente el comportamiento que queríamos representar.

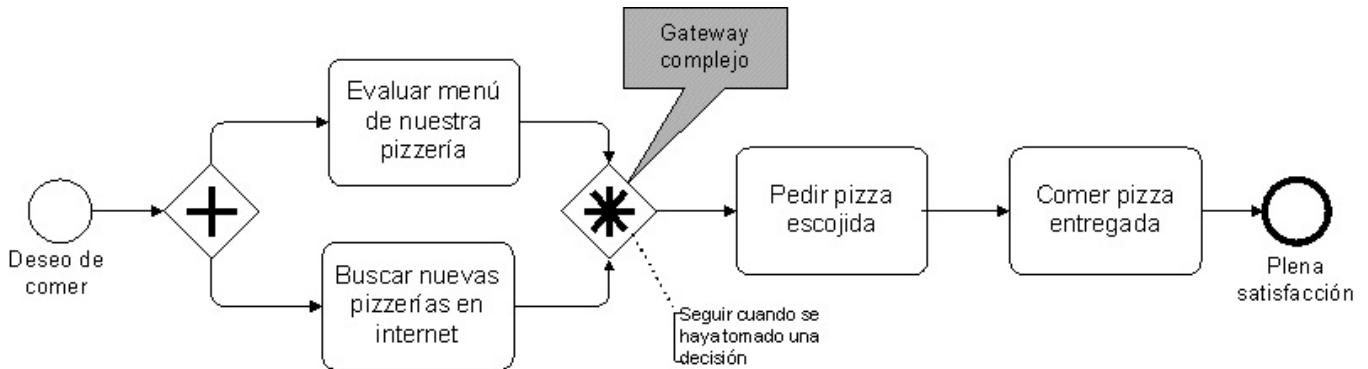


Figura 2.19: Pedido de pizza con un Complex-Join

El otro escenario representado en la figura 2.20 es parecido. Vamos a suponer que vamos a ejecutar cuatro actividades en forma simultánea. La quinta actividad debe ejecutarse apenas se disponga del resultado de tres actividades, independiente de algún orden correlativo. Este tipo de comportamiento de sincronización se puede representar con un Gateway complejo, agregándole si la condición en forma de un artefacto del tipo comentario.

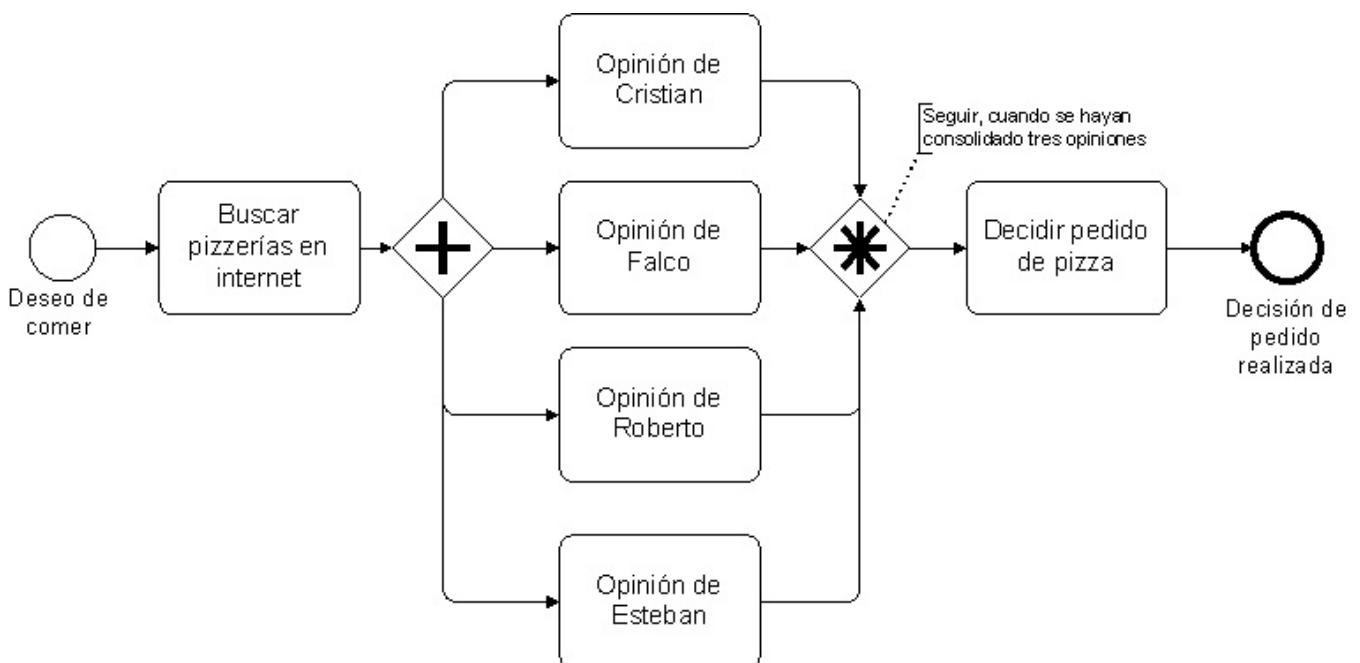


Figura 2.20: Con Gateway complejos se pueden representar también casos como «M out of N-Joins»

En un principio también podemos utilizar el Gateway complejo del tipo Split, por ejemplo para evitar el uso de varios Gateways por separado y hacer más compacto el diagrama. En el caso del OR-Split que muestra la figura 2.14 podría reemplazarse por un Gateway complejo escribiendo un comentario en el símbolo del Complex-Split. Nosotros no consideramos adecuado el uso del caso del complexsplit, no lo hemos utilizado nunca en la práctica ni tampoco nos hemos encontrado con algún modelo hecho por otros profesionales que lo hayan utilizado.

2.4. Flujo de Procesos sin Gateways

Algunas personas no les gustan los Gateways, porque opinan que las compuertas aumentan demasiado la amplitud de los diagramas y preguntan si no existe una forma más compacta para la notación. Los autores de la notación de BPMN le han dado en el gusto a esta gente y han considerado para ciertos casos (simples) poder desistir de los símbolos de Gateways (XOR-, AND- y OR-Gateways) y reemplazarlos por «flujos condicionales» que se asocian directamente a las actividades. Si usted es parte de esta comunidad de personas que no le gustan los símbolos de Gateways puede modelar con flujos condicionales, pero debe hacerlo con mucha más cautela y probar bien el comportamiento y la semántica, porque se ha visto en algunos casos que este tipo de notación no puede representar todos los patrones y el modelador debe recurrir a los Gateways. Analicemos algunos casos.

En la figura 2.21 mostramos el diseño de alternativa al OR-Split como el XORJoin. El modelo superior e inferior muestran el mismo tipo de patrón de workflow. El XOR-Join dirige directamente el o los token a la actividad 4, es decir en el caso superior se desiste de la compuerta y se asocian directamente los flujos secuenciales a la actividad 4, lo cual hasta el momento no nos genera problemas de interpretación. En el caso del OR-Split podemos reemplazar la compuerta por un elemento de flujo llamado «flujo condicional», que lo reconocemos por un pequeño rombo al inicio del flujo. Este símbolo puede sólo ser asociado en la salida de actividades (o subprocessos), pero no a Gateways o eventos.

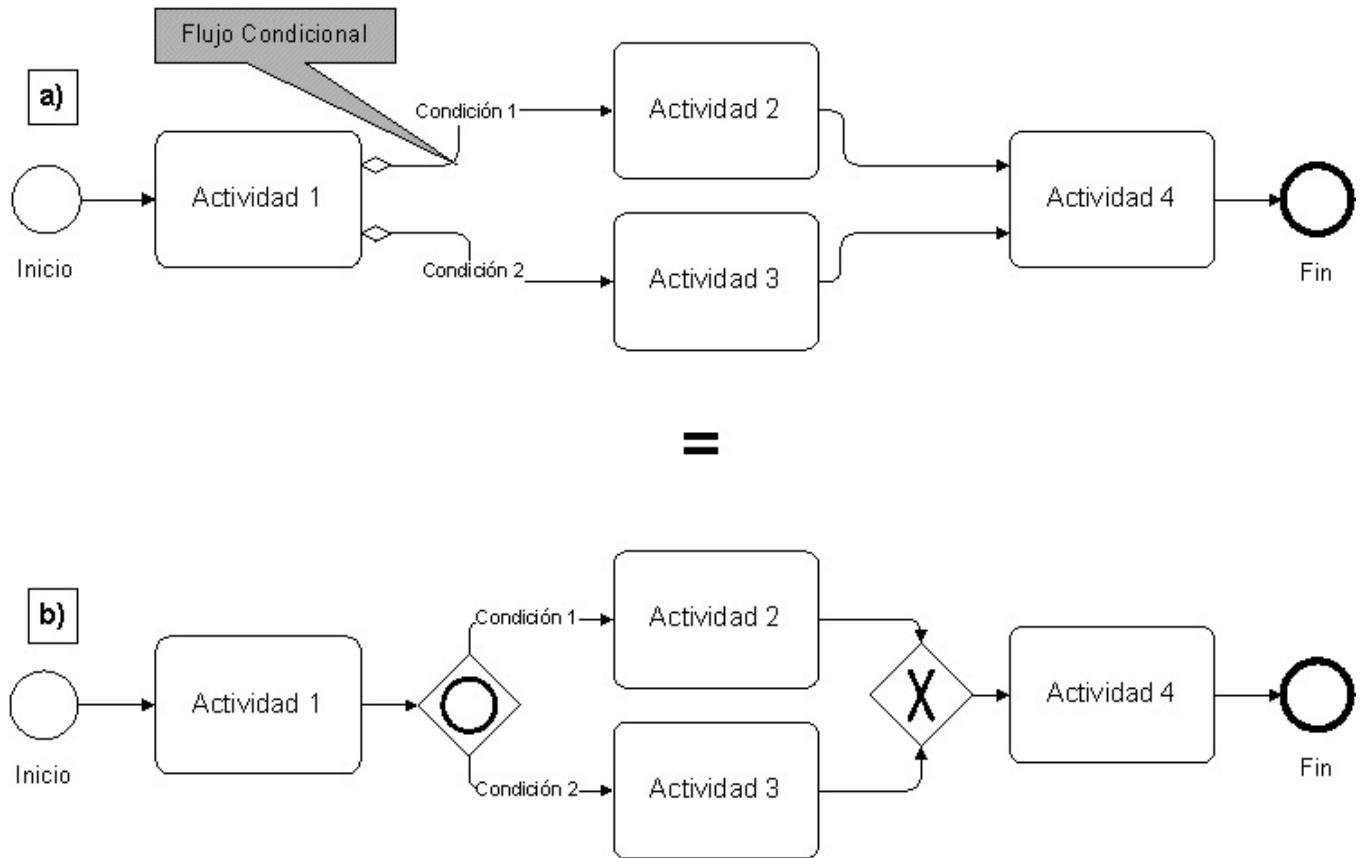


Figura 2.21: Equivalencia entre a) utilización de flujos condicionales y b) utilización de OR-Split y XOR-Join

¿Qué le parece esta notación alternativa? Si usted leyó detenidamente la sección 2.3 debería haber reparado en los siguiente: Si es válida una sola condición, no tenemos problemas, pero si ambas son válidas el OR-Split genera dos tokens que gracias al XOR-Join ejecutan dos veces la actividad 4. Este comportamiento no tiene que ser necesariamente falso, pero tampoco deseado y esto nos lleva al primer problema, que está relacionado con la posibilidad de desistir de la simbología de Gateways:

Sin Gateways no podemos modelar sincronizaciones (AND-Joins).

El segundo problema es la insuficiencia de poder representar combinaciones de validación de condiciones. La lógica de proceso representada en la figura 2.22, no permite ser modelado sin Gateways debido a la necesidad de consultar el estado por medio de un evento intermedio.

Recuerde: flujos condicionales no se pueden asociar a eventos.

El tercer problema consiste en el hecho que flujos condicionales obedecen a la misma lógica que el OR-Split, es decir que las condiciones de los flujos condicionales no pueden representar el caso de exclusión. Desde el punto de vista técnico, este hecho no es relevante porque el OR-Split es compatible con el XOR-Split (compatible hacia abajo), pero el lector debe asumir o más bien tener claro que este comportamiento puede ser así. Nuestra experiencia nos ha mostrado que modelos diseñados con flujos condicionales, llevan rápidamente a confusiones en su interpretación.

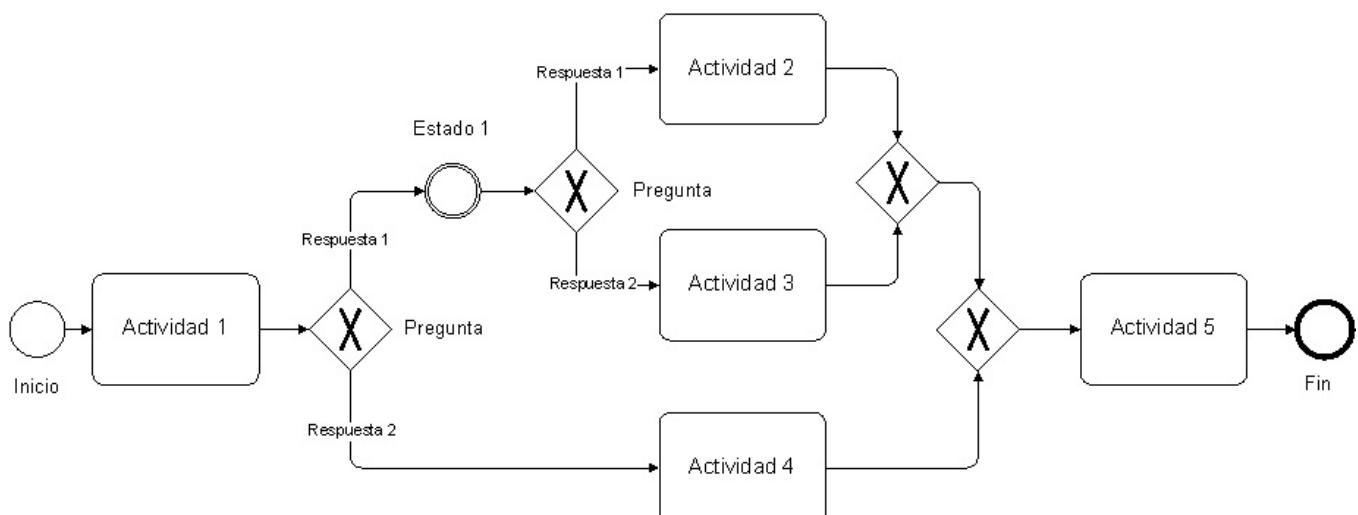


Figura 2.22: XOR-Gateways combinados

¿Deberíamos entonces desistir en forma general de usar flujos condicionales? En realidad no deberíamos ser tan categórico. En procesos sencillos los Gateways pueden hacer parecer los modelos más complejos de lo que son. Por ejemplo loops sencillos son más expresivos usando un flujo condicional que interponiendo dos Gateways (ver figura 2.23). BPMN nos permite además de asociar varios flujos de salida a los eventos de inicio y hacer converger varios flujos a los eventos de

término. En la figura 2.23 vemos un mismo proceso modelado con y sin Gateways, pero en el detalle los modelos no son exactamente iguales. En el modelo superior el XOR-Gateway excluye sintácticamente, que puedan ser recorridos varios flujos, es decir condición 1 y 2 nunca podrían ocurrir en forma simultánea. En el modelo sin Gateways, la sintáctica implícita del flujo condicional no puede excluir la ocurrencia de ambas condiciones.

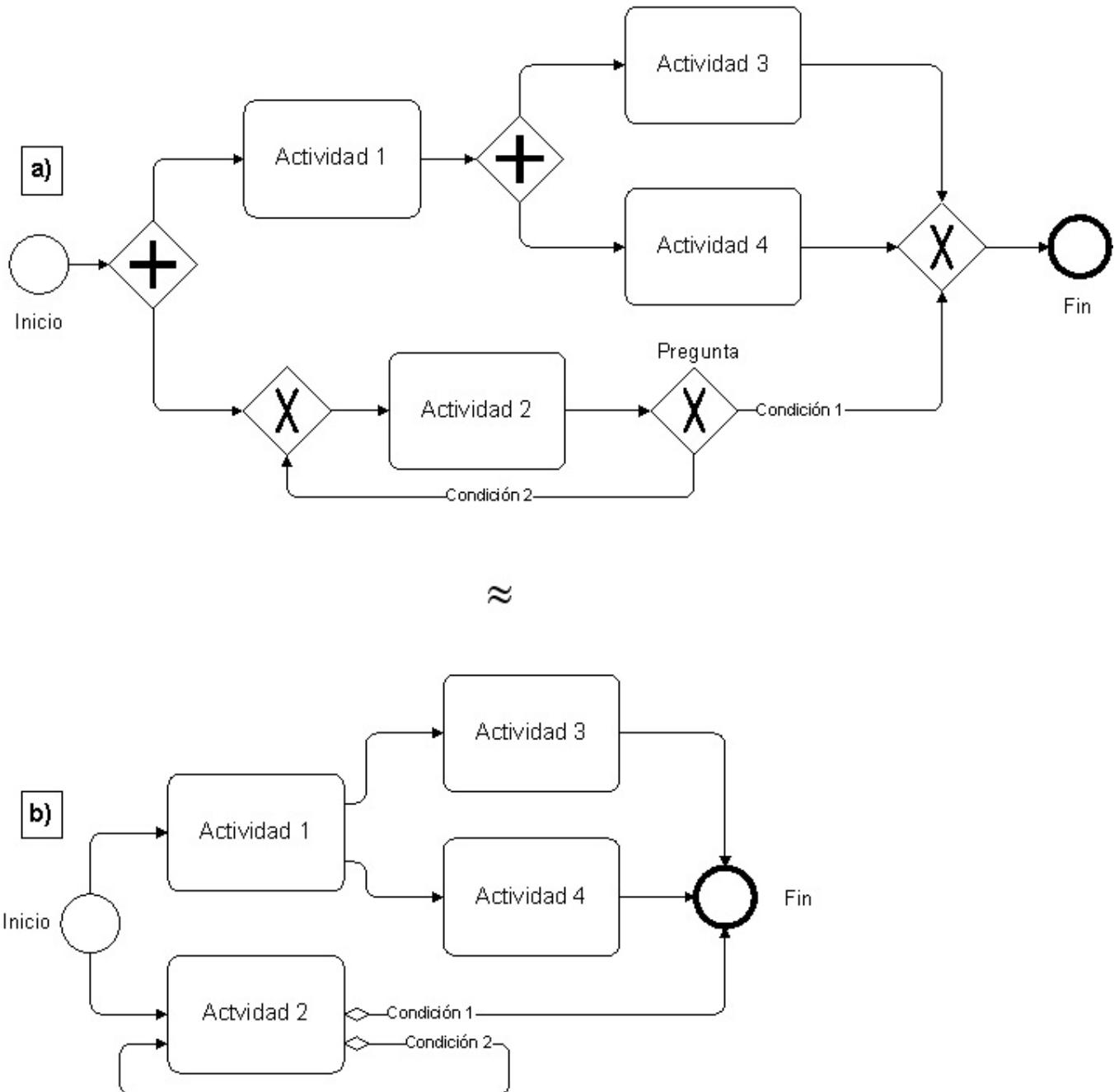


Figura 2.23: Similitud entre a) utilización de AND-Gateway y XOR-Gateway b) utilización de flujos condicionales

2.5. Lanes

Hasta el momento hemos hablado sólo como fluye el proceso o «qué» es lo que hay

que hacer en el proceso, pero aún no hemos visto quién o quienes son los responsables de ejecutar las actividades. BPMN utiliza carriles llamados «lanes» para la asignación de responsables.

En la figura 2.24 vemos en nuestro ejemplo que a las actividades se le han asignado nombres de personas. De acuerdo a esta nomenclatura podemos deducir que si Cristian tiene deseo de comer, selecciona una receta en su libro de cocina. De acuerdo al modelo presentado, él sólo cocina pasta, pero si quiere freír steaks, tendría que invitar a otro amigo para que cocine. Según el modelo, al término del proceso de cocina, Cristian puede comerse la preparación. Las tres personas que aparecen en este modelo comparten este proceso en una residencia, quienes al mismo tiempo representan los posibles participantes del proceso descrito. En BPMN se les denomina a estos participantes «pools». Los pools los vamos a tratar en forma separada en la sección 2.9.

En este ejemplo asignamos lanes a personas, pero la especificación de BPMN no hace ningún tipo de clasificación al respecto. Por lo general se usan lanes en la práctica para:

Cargo de algún área (como gerente, supervisor, etc.) Roles (como auditor, usuario de negocio, ejecutivo, etc.) Roles generalizados (como cliente, proveedor, fiscalizador, etc.)

Departamentos (como contabilidad, ventas, etc.) Aplicaciones o sistemas (como CRM, SAP, etc.)

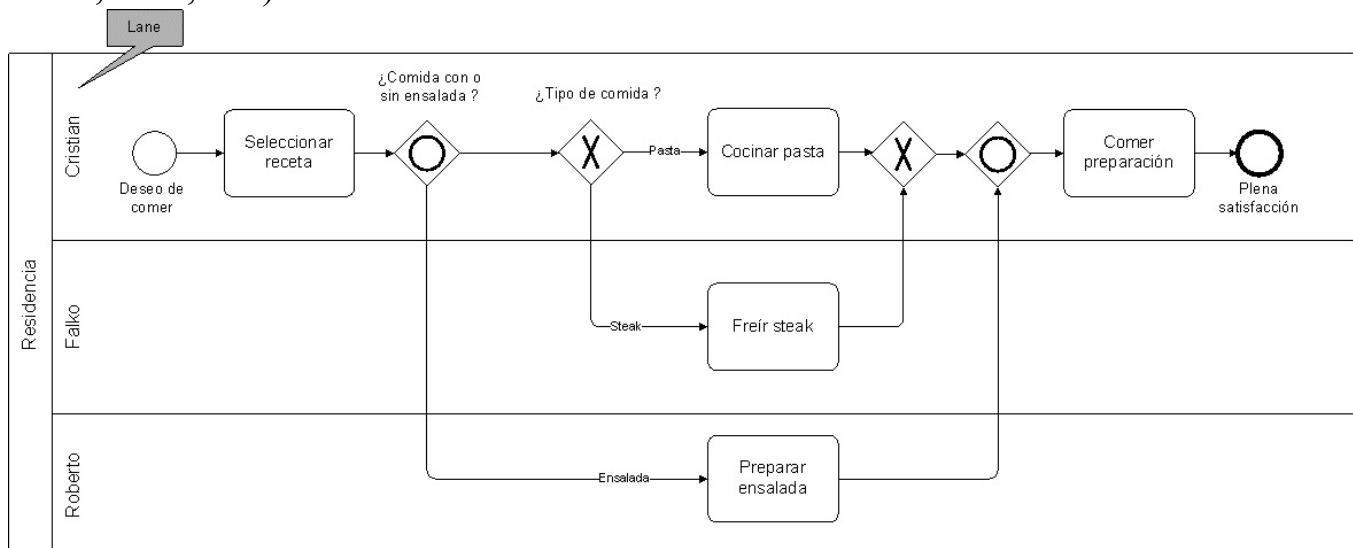


Figura 2.24: Utilización de Lanes

El término «lane» tiene su historia en las notaciones de modelamiento de procesos. Muchas notaciones se han apoyado de esta forma de representar participantes en los modelos de procesos. El término se utilizó en analogía a los carriles en las piscinas de natación, en el cual cada nadador tiene su propio carril, separado por las bandas de boyas flotantes. En la versión anterior a BPMN 2.0 también se podía estructurar

una jerarquía de lanes dentro de un pool como lo muestra la figura 2.25.

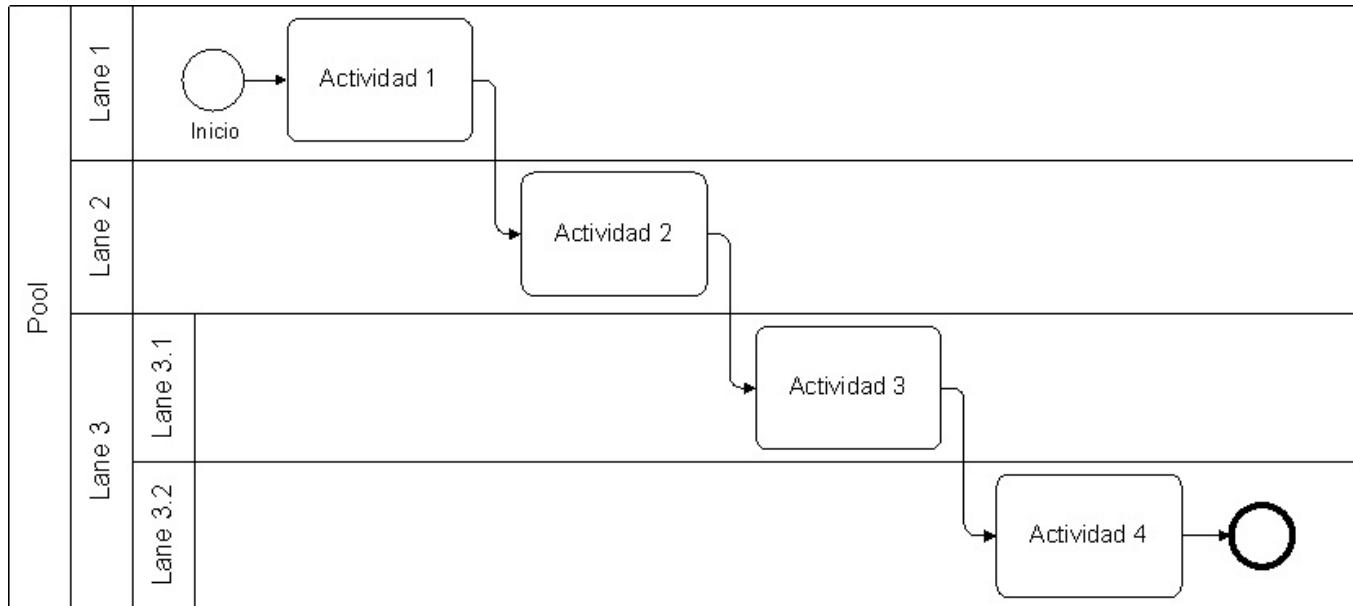


Figura 2.25: Estructura jerárquica de lanes

En la práctica a veces no es fácil encontrar la estructura adecuada de lanes. Seguramente en nuestro ejemplo reciente ya se dio cuenta, que Cristian invita a cocinar pero no a comer, pero el modelo final que queremos representar es que todos los participantes vayan a cenar juntos. Según las reglas de BPMN, un objeto de flujo (actividad, evento, Gateway) sólo se puede posicionar dentro de un lane y no entre ellos.

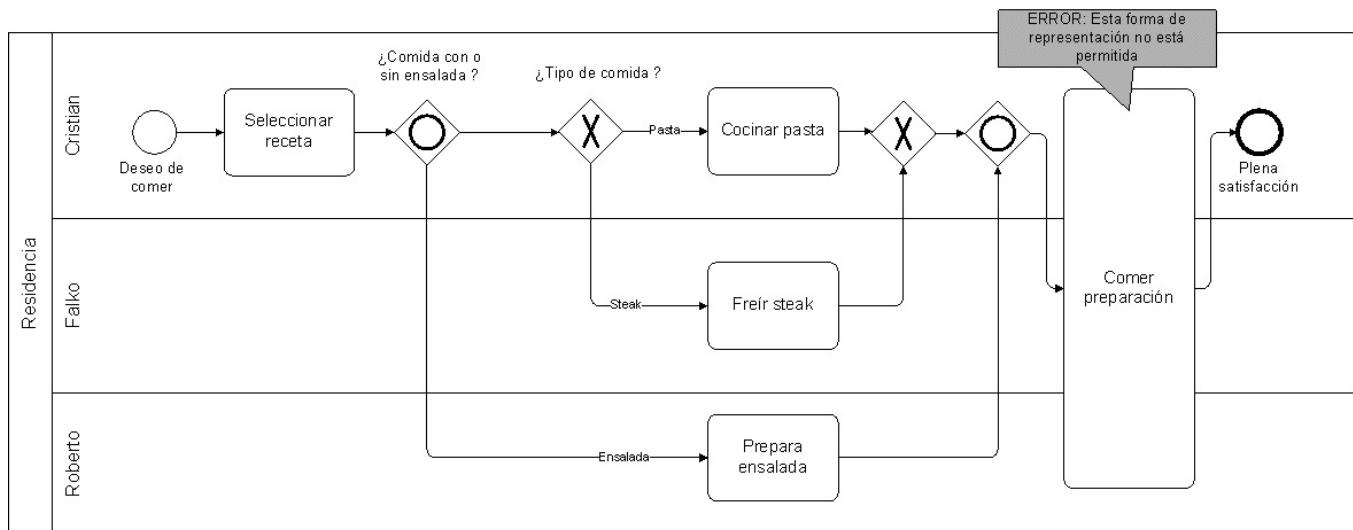


Figura 2.26: Uso incorrecto de actividad en Lanes

La solución para este escenario, sería crear la actividad tantas veces como personas participan en la cena (ver figura 2.27). Esta forma de representación corresponde también con la realidad, porque cada persona come en su plato, pudiendo terminar unos antes que otros.

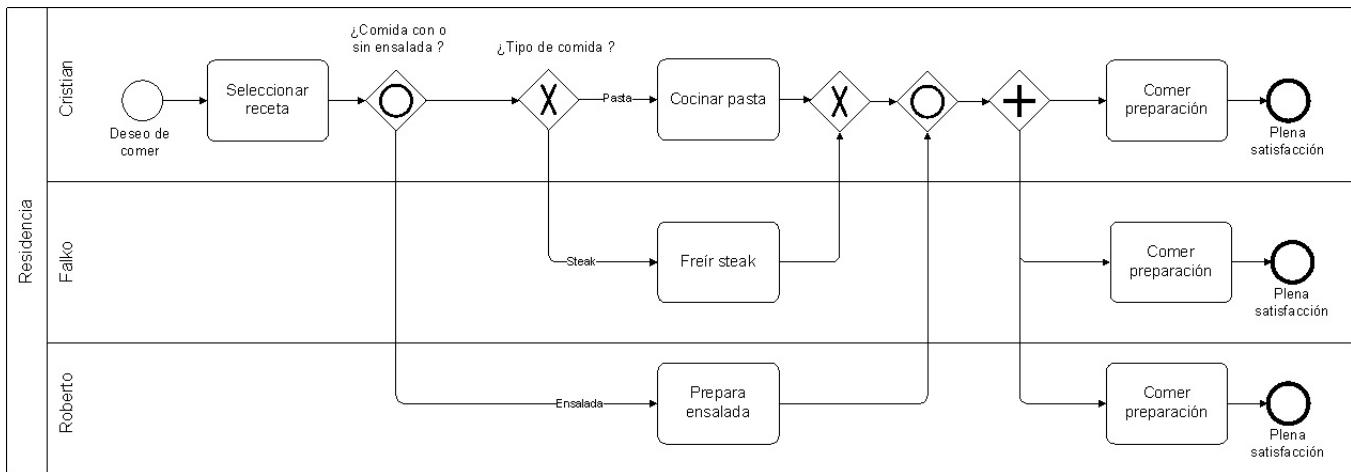


Figura 2.27: Uso correcto de actividad en lanes

Qué hacer o cómo representar cuando un grupo de personas trabaja en una tarea en conjunto, como es el caso de las comisiones, comités, directorios etc. Para estos casos se recomienda crear un lane propio con el nombre del grupo.

Cabe recordar que los diagramas y ejemplos de lanes presentados en este libro corresponden a la nueva especificación de BPMN 2.0 en donde ya no se permite la jerarquización de lanes ni tampoco una linea separatoria (vertical) (ver figura 2.28).

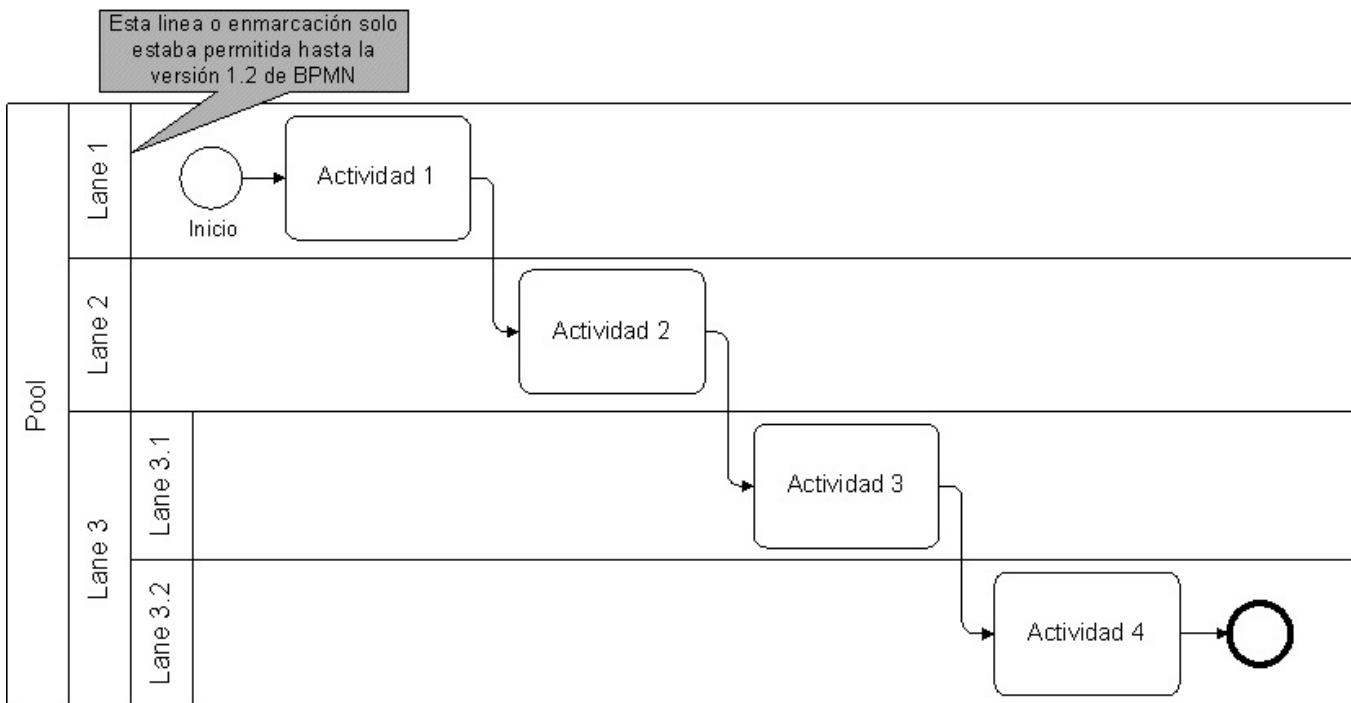


Figura 2.28: Estructura de lane prohibida a partir de la versión BPMN 2.0

2.6. Eventos 2.6.1. Importancia en BPMN

Con las actividades y los Gateways conocimos dos de los tres elementos de flujos en BPMN: Las tareas (actividades) cambian el estado de un objeto, bajo ciertas condiciones (Gateways). Uno de los elementos principales de flujo que nos falta

conocer son cosas que ocurren (eventos). Estos elementos de flujo no son de menor importancia que las actividades o los Gateways en BPMN, razón por la cual le vamos a dedicar una sección completa a este elemento estructural de la notación.

En la sección 2.2 ya describimos algunas características de los eventos: Eventos de captura y eventos disparadores

Eventos de inicio, eventos intermedios y eventos de término

Los eventos de captura se les denomina en BPMN «catching events» e indican ocurrencias que vienen de afuera y a los que un proceso debe reaccionar cuando estos suceden, independiente si este tipo de eventos «gatillan» (en inglés: trigger) el inicio de un proceso o suceden durante un proceso. Lo importante de entender es que este tipo de sucesos tienen un impacto sobre el proceso y que este debe reaccionar. Eventos de captura pueden tener el siguiente impacto sobre los procesos:

Inician el proceso,
el proceso o el flujo del proceso continúa,

una actividad o un subprocesso que se encuentra en ejecución es interrumpida o cancelada,

durante la ejecución de una actividad o un subprocesso, impulsa el inicio de otra actividad o subprocesso (nuevo en BPMN 2.0).

Los eventos del tipo disparador se les denomina en BPMN «throwing events» e indican eventos creados dentro del proceso. Es decir a diferencia a los eventos del tipo de captura a los cuales el proceso debe reaccionar, en este caso el mismo proceso actúa como gatillador de nuevos eventos. Eventos del tipo disparador:

pueden crearse durante el proceso, o
al final del proceso (eventos de término).

Podemos clasificar entonces todos los eventos de inicio como eventos de captura. Sería contradictorio si un proceso pudiese crear un evento antes que haya sido iniciado. La forma más sencilla de representar un evento de inicio lo podemos apreciar en la figura 2.29. Apenas ocurra el evento, el proceso se inicia.



Figura 2.29: Un Evento gatillante de proceso

Cabe recordar que el signo de interrogación indica que este evento podría obedecer a una clasificación determinada por BPMN. Hasta el momento hemos conocido sólo los eventos del tipo indefinido, pero más adelante vamos a conocer como BPMN

clasifica eventos.

También es posible que varios eventos puedan iniciar un proceso, como lo muestra la figura 2.30. Es importante entender que cada evento inicia una nueva instancia de proceso.

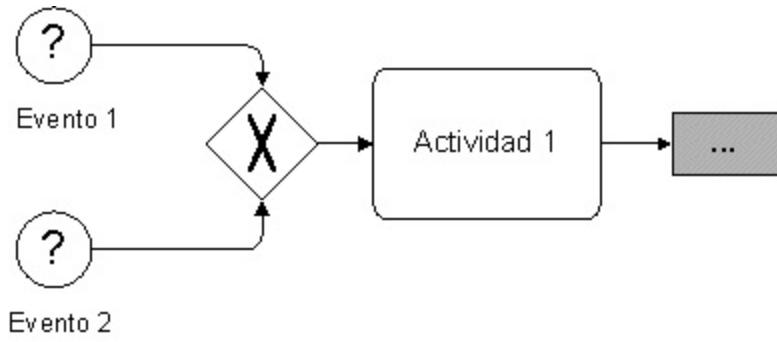


Figura 2.30: Dos eventos que independientemente pueden gatillar un proceso (evento 1 o evento 2)

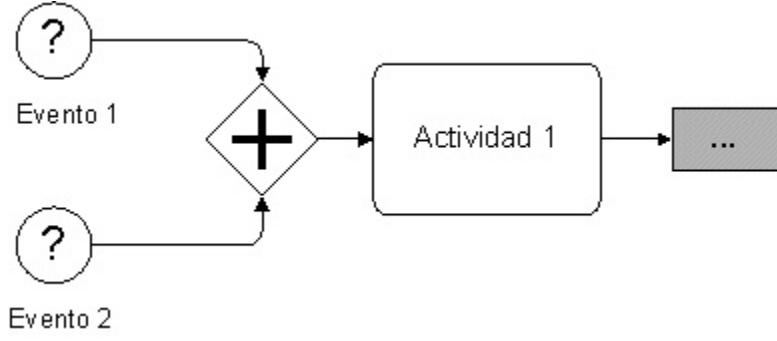


Figura 2.31: Dos eventos que producirían un punto muerto (deadlock) en interpretadores de BPMN

Si quisieramos modelar el caso que varios eventos tienen que suceder para que pueda iniciarse el proceso, muchos analistas lo representarían en forma intuitiva como muestra la figura 2.31. Desgraciadamente esta no es la forma verdaderamente correcta de representar este patrón en BPMN, a pesar que al lector le cueste comprenderlo. La razón de esto es que el AND-Join explicado en la sección 2.3.2 no apoya la correlación necesaria, para que ambos sucesos sean reconocidos como un par. Vamos a explicar y dar una solución a este problema en la sección 2.6.14, y describir la propuesta de la versión BPMN 2.0.

Para que un proceso que está en espera pueda continuar, puede hacerse necesario que ocurra un evento intermedio, como lo muestra la figura 2.32. El flujo se interpreta de la siguiente forma: Si la actividad 1 ha terminado, tiene que ocurrir primero el evento 1, antes que puede iniciarse la actividad 2. De acuerdo a nuestro concepto de token, espera el token tanto tiempo en el evento 1 hasta que ocurra y sólo al finalizar el token sigue su camino y gatilla el inicio de la actividad 2. El evento 1 de la figura 2.32 pertenece a la categoría de eventos del tipo disparador y no a los de captura.



Figura 2.32:

Utilización de evento intermedio

Pero, ¿cómo podemos ahora representar que un proceso tiene que esperar la ocurrencia de dos eventos? No se le ocurra modelarlo como muestra la figura 2.33. ¿Por qué? Cuando la actividad 1 haya concluido, el token avanza hacia el evento 1 y espera que este ocurra. Supongamos ahora que el evento 2 sucede antes que el evento 1, entonces al ocurrir más tarde el evento 1, el token avanza al evento 2 y espera que este suceda, a pesar que ya ocurrió. Como el evento 2 ya se «esfumó», el token quedará indefinidamente esperando que este suceda, lo que técnicamente produciría un deadlock. Podríamos admitir que usuarios de negocio lo interpretarían bien, si modelamos así el caso, pero técnicamente el concepto de token no lo puede interceptar, por lo que se haría necesario cambiar el diseño en el modelo técnico (ver sección 5.4.2).

Si necesitamos modelar el caso de tener que esperar la llegada de dos eventos que suceden en forma independiente, y cuya condición es necesaria para que continúe el proceso, lo modelamos como muestra la figura 2.34.

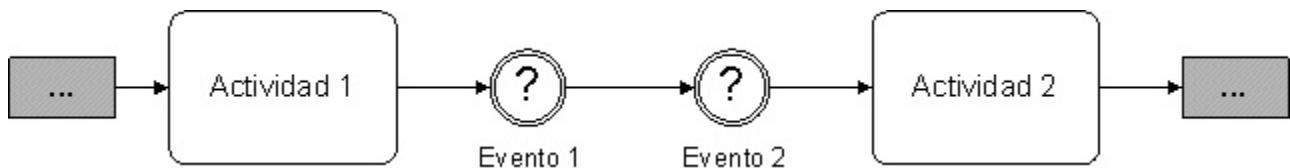


Figura 2.33: Utilización de dos eventos intermedios continuos

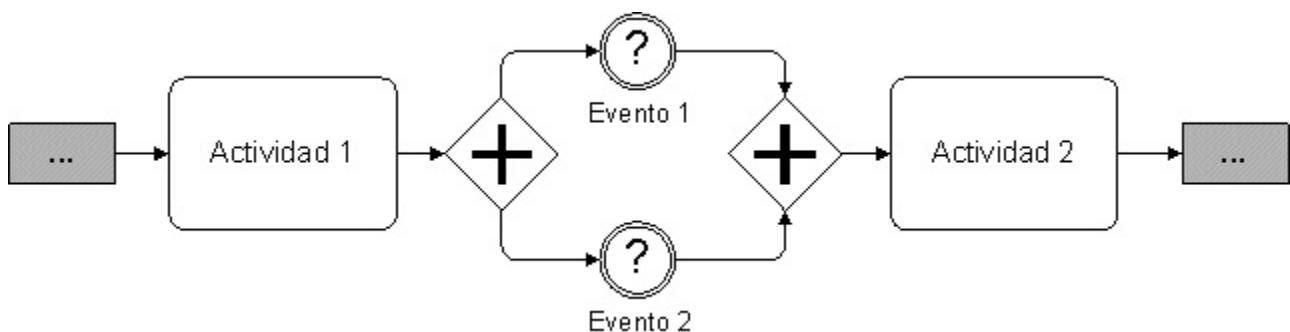


Figura 2.34: Utilización de AND-Gateway con eventos intermedios

En BPMN también podemos representar casos en que existe un flujo normal, pero puede ocurrir un evento inesperado que interrumpa una actividad o un subproceso. A estos eventos intermedios se les llama «sobrepuertos» (en inglés: attached), ya que van superpuestos a un costado de la actividad (ver figura 2.35). Si sucede un evento de este tipo, interrumpe la marcha de la actividad, cualesquiera que sea su estado intermedio. En este caso el token tendría el siguiente comportamiento:

Primero avanza hacia la actividad 1, la cual se inicia.

Si sucede el evento 1, durante la ejecución de la actividad 1, esta se interrumpe inmediatamente y el token sigue su flujo en la actividad 3 (caso de excepción). Si no sucede el evento 1, la actividad 1 se ejecuta en forma normal y el token sigue su flujo regular e inicia la actividad 2.

Si sucede el evento 1, después que se haya ejecutado la actividad 1, el suceso no impacta en el proceso.

En BPMN hasta la versión 1.2 los eventos sobrepuertos (a excepción de los eventos del tipo de compensación), siempre cancelan la actividad en ejecución. Este comportamiento no siempre refleja la realidad, por lo que en la versión 2.0 se introdujo un nuevo símbolo: un evento intermedio sobrepuerto, pero del tipo «no interrupción». En este caso el token tiene otro comportamiento como muestra la figura 2.36:

El token inicia, como en el caso anterior, la actividad 1.

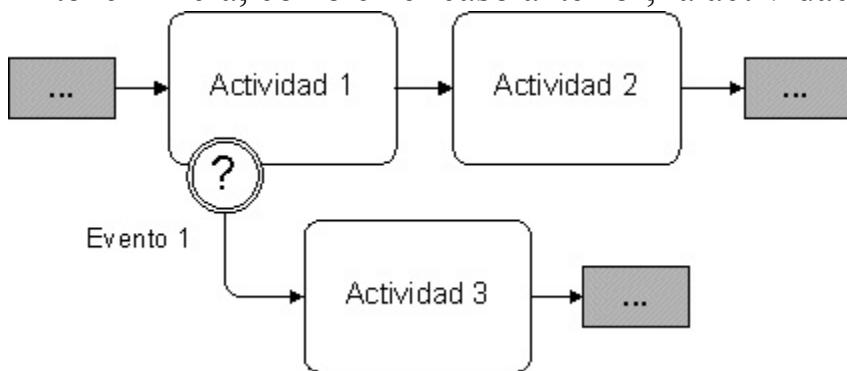


Figura 2.35: Utilización de

evento intermedio y sobrepuerto (attached)

Si sucede el evento 1, durante la ejecución de la actividad 1, el token es clonado: La actividad 1 sigue en proceso, pero al mismo tiempo avanza el segundo token (el clonado) a la actividad 3, la cual también es iniciada y ejecutada. Este evento puede suceder incluso en forma repetitiva, y el token vuelve a clonarse hasta que la actividad 1 haya terminado.

Si no sucede el evento 1, la actividad 1 se ejecuta en forma normal y el token sigue su flujo regular e inicia la actividad 2.

Si sucede el evento 1, después que se haya ejecutado la actividad 1, el suceso no impacta en el proceso, igual que en el caso anterior.

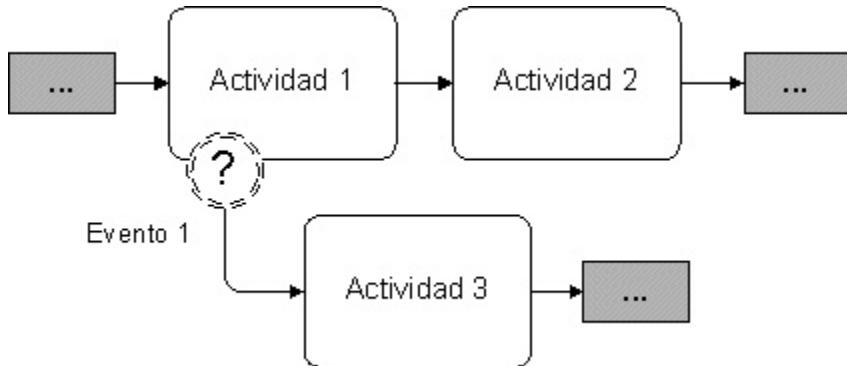


Figura 2.36: Utilización de evento intermedio y sobrepuerto del tipo "no interrupción"

Eventos intermedios del tipo «disparador» son gatillados por el proceso mismo, es decir un token que llega a un evento de este tipo, lo gatilla y avanza inmediatamente al siguiente elemento en el flujo. Eventos del tipo disparador, no interrumpen una actividad, razón por la cual nunca se deben sobreponer en un objeto, sino que deben usarse asociados al flujo de secuencia. Hasta ahora conocemos el evento intermedio del tipo indefinido, con el cual podemos indicar que hemos alcanzado un estado. También este caso es un evento del tipo disparador.

En las próximas secciones vamos a presentar diferentes tipos de eventos que se pueden utilizar en BPMN:

- Eventos de mensaje
- Eventos de tiempo
- Eventos de error
- Eventos de condición
- Eventos de señal
- Eventos de término
- Eventos de conexión (link)
- Eventos de compensación
- Eventos múltiples
- Eventos de cancelación

También vamos a tratar bajo la rúbrica de eventos, los Gateways basados en eventos, los cuales reaccionan sólo ante la ocurrencia de eventos y otras novedades que trae la versión BPMN 2.0.

2.6.2. Evento de mensaje

La mayoría de los procesos requieren tarde o temprano de una comunicación. En BPMN los eventos que portan información se les llama «eventos de mensajes» y se les reconoce por un símbolo de sobre de carta. Las variantes posibles de uso se encuentran en la figura 2.37. En principio el término «mensaje» no se restringe a ciertos portadores de información como cartas, emails o llamadas, sino a cualquier

objeto que porte información. Por ejemplo en la figura 2.38 modelamos el pedido de pizza con dos eventos intermedios que portan información, la información del pedido (orden de compra) y la información de la entrega (guía de despacho, boleta o factura).

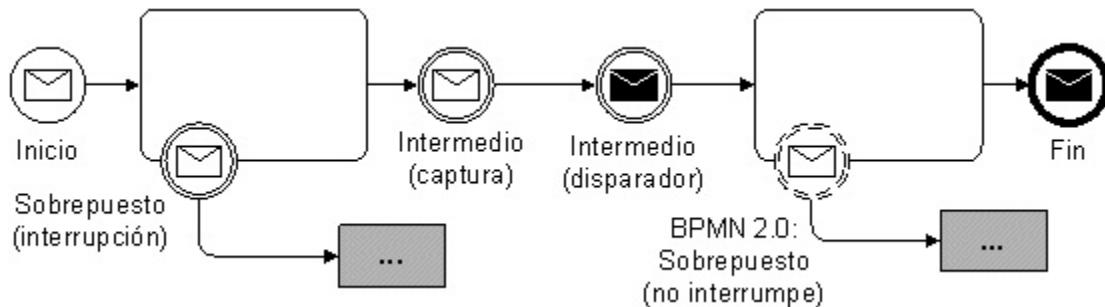
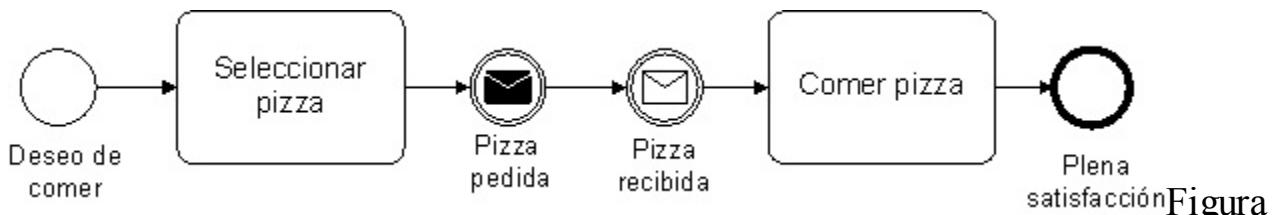


Figura 2.37: Tipos de eventos de mensajería En el ejemplo de la figura 2.38 usted puede apreciar que desistimos de agregar una actividad «pedir pizza». Muchos analistas aún modelan el caso como muestra la figura 2.39, lo que según la especificación no es correcto. El evento intermedio «pizza pedida» (del tipo disparador) implica que la pizza haya sido pedida, dicho de otra forma en BPMN los eventos disparadores asumen una acción y una ocurrencia al mismo tiempo. En el ejemplo de la figura 2.39 la acción del pedido de pizza está definida dos veces.



2.38: Pedido y recepción de pizza como eventos de mensaje

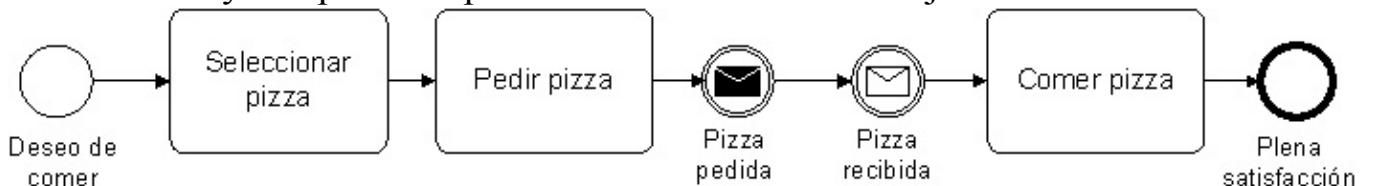


Figura 2.39: Caso de incorrecta utilización de actividad con evento intermedio

Un evento de mensaje también puede interrumpir una operación como lo muestra la figura 2.40. En este ejemplo asumimos el rol de operador de alguna aplicación web. El cliente nos llama haciendo un reclamo que el sitio no responde o está abajo. El operador busca el problema o verifica si existe algún error. Es posible que el cliente se haya equivocado y el problema lo tenga él, porque en algún instante se le haya caído internet. Él llama nuevamente para que no nos preocupemos. Este llamado entra al proceso como un evento de mensaje del tipo interrupción, al ocurrir todas las demás actividades se cancelan y el flujo sigue a la actividad asignada por el evento de interrupción.

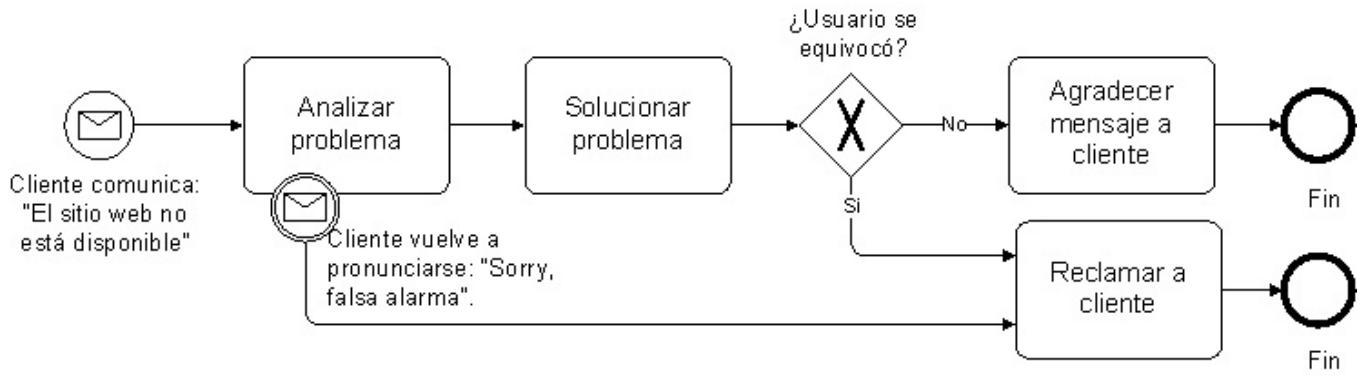


Figura 2.40: Utilización de evento intermedio y sobrepuerto del tipo "mensaje"

Nuestro Tip en BPMN

Algunas personas no se pueden familiarizar con el evento intermedio del tipo disparador, el hecho que además asume una acción (enviar mensaje), puede confundir al lector. Muchas veces BPMN permite modelar una situación determinada de diferentes formas sin cometer errores sintácticos. En este caso podemos desistir del evento del tipo disparador y reemplazarlo por una actividad de «envío», como sería el caso de la figura 2.41 con la actividad «Pedir pizza». En la sección 2.7 vamos a conocer tipos de actividades especializadas en el envío y recepción de mensajes.

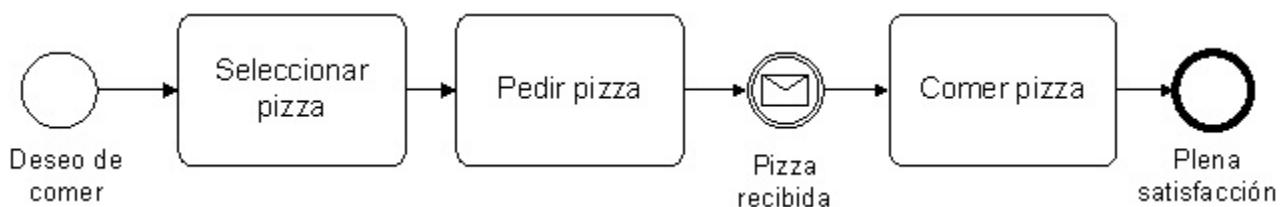


Figura 2.41: Caso de correcta utilización de Actividad (para envío) y Evento (para recepción)

2.6.3. Evento de tiempo

El evento de tiempo (también llamado temporizador) se utiliza cuando una condición de tiempo ocurre. Gráficamente se reconoce como un reloj como marcador dentro del evento. La variantes posibles de uso se encuentran en la figura 2.42. Como evento de inicio se puede utilizar para:

- iniciar cada ciertos intervalos un proceso,
- iniciar un proceso regularmente en una fecha y hora indicada,
- iniciar un proceso en una relación temporal con otro evento e
- iniciar un proceso por única vez en una fecha y hora determinada.

Como evento intermedio el temporizador puede detener el proceso, hasta que:

- un tiempo definido se haya alcanzado,
- un período de tiempo haya transcurrido,
- se haya alcanzado un tiempo, que se encuentre en relación a otro evento.

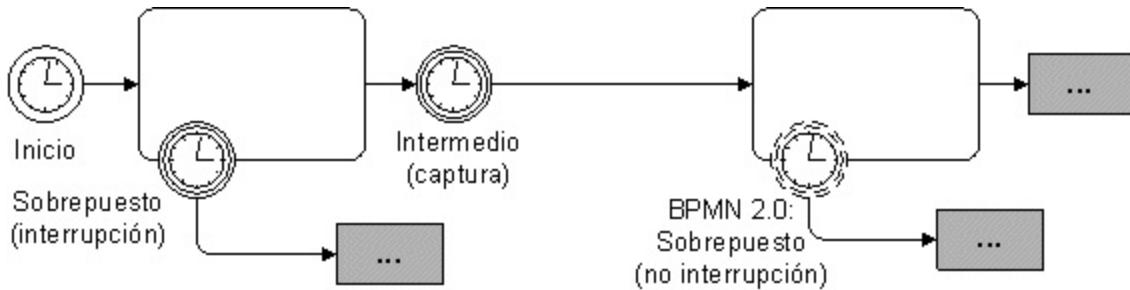


Figura 2.42:

Tipos de Eventos de Tiempo

En la figura 2.43 se aprecian algunos ejemplos de uso del temporizador. Un evento de tiempo no puede ser impulsado por un proceso, porque sobre el tiempo no tenemos influencia, razón por la cual este evento existe sólo en forma de «evento de captura».

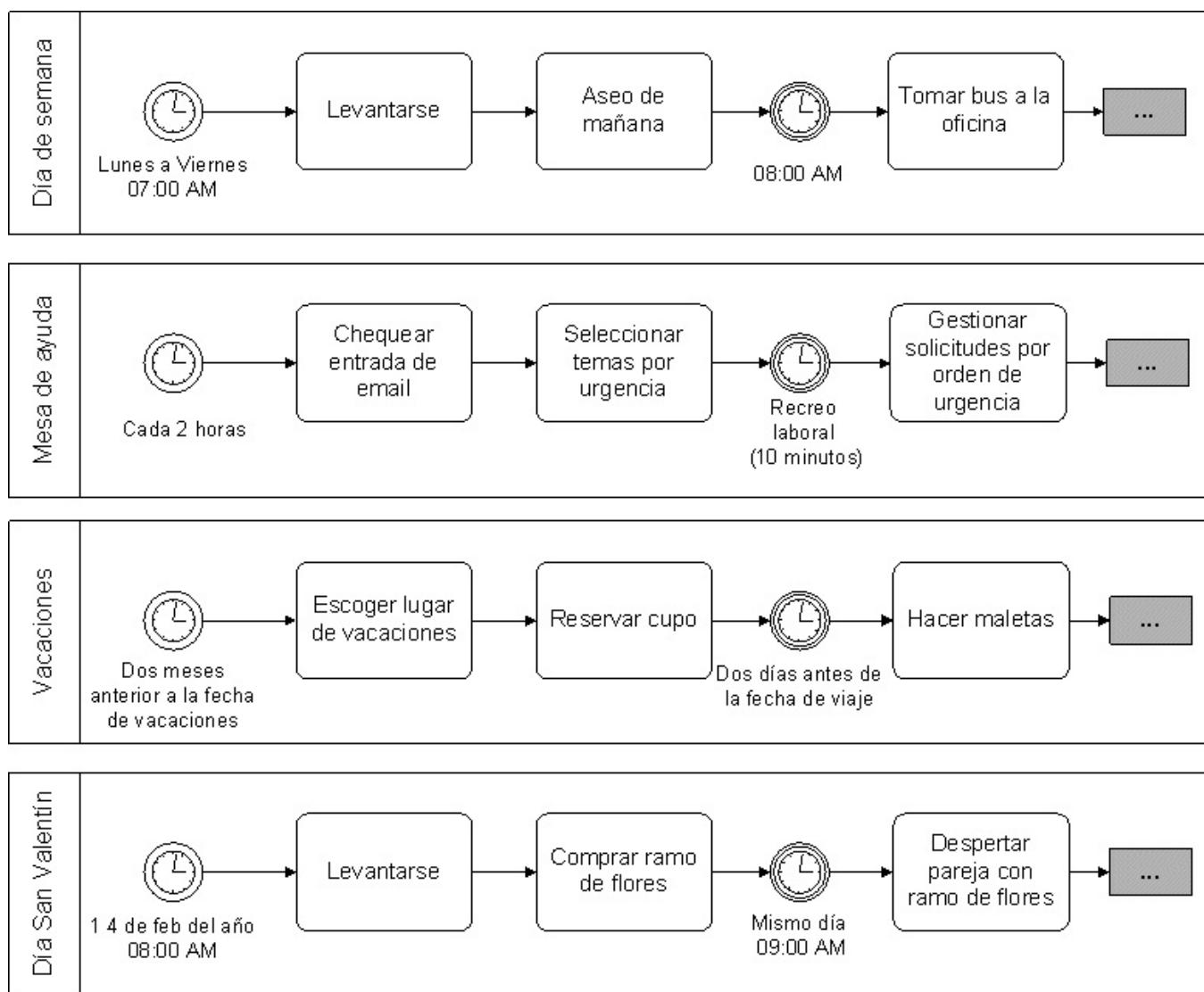


Figura 2.43: Utilización de eventos de tiempo

Muy a menudo se utiliza el temporizador sobrepuuesto como «timeout», tiempo máximo permitido para la ejecución de una actividad. En la figura 2.44 mostramos un ejemplo donde sólo nos damos un tiempo de 30 minutos para escoger una pizza.

Si no nos ponemos de acuerdo dentro este lapso, cancelamos la búsqueda y cocinamos pasta, pero en ambos casos vamos a comer lo que llegue a la mesa.

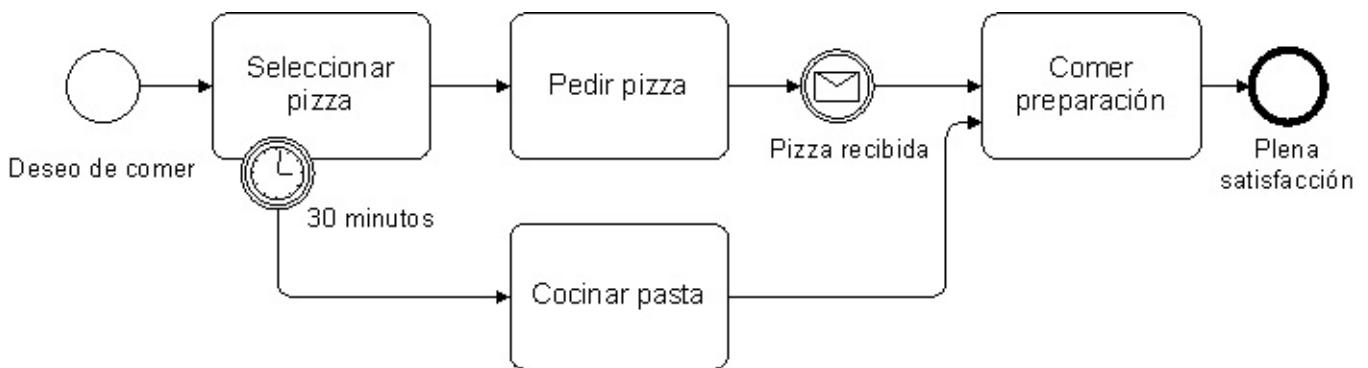


Figura 2.44: El timeout para la actividad seleccionar pizza es de 30 minutos

A partir de la versión 2.0 en BPMN también se pueden utilizar temporizadores sobrepuertos que no interrumpen la actividad. Un ejemplo para un evento del tipo temporizador sin interrupción lo vemos en la figura 2.45. Antes de poder cenar juntos hay que preparar la comida y poner la mesa, que lo haremos sólo 10 minutos antes que esté lista la comida.

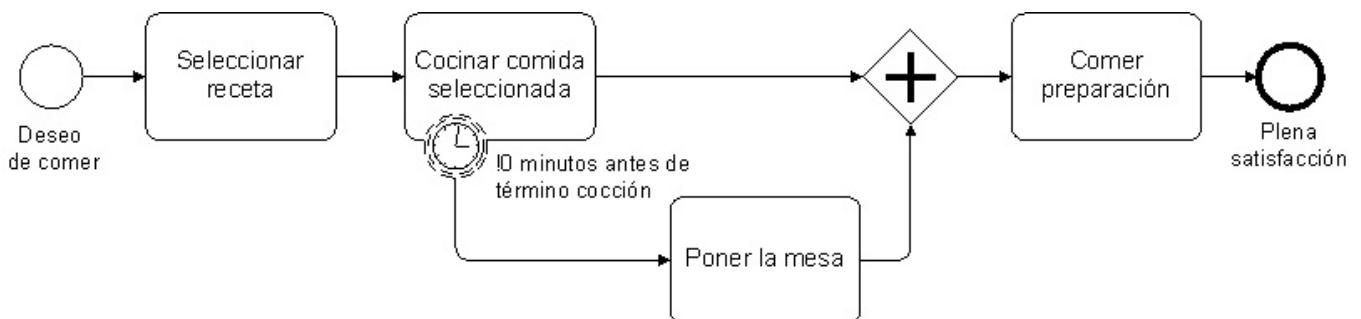


Figura 2.45: Utilización de evento de tiempo, intermedio y sobrepuerto del tipo "no interrupción"

2.6.4. Evento de error

En pocas ocasiones se puede asumir que durante la ejecución de los procesos no ocurren errores. Si usted identifica los puntos donde pueden ocurrir errores, los puede interceptar utilizando este tipo de eventos. Gráficamente se representan con un signo de rayo dentro del círculo del evento como lo muestra la figura 2.46.

La notación de BPMN no especifica tipos de errores. Debido a lo anterior vamos a darles en la sección 4.5.1 algunas recomendaciones prácticas al respecto.

Sin embargo en BPMN se considera un error como un evento excepcional, razón por la cual sólo se puede modelar como evento intermedio sobrepuerto y que además requiere de un tratamiento excepcional. Como tipo disparador sólo se debe usar como evento final, indicando que el proceso ha sido cancelado por error, o bien el evento es capturado por un subproceso superior que lo lleva a un tratamiento especial. La colaboración entre subprocesos (superior vs inferior) se explicará en la

sección 2.8. Allí encontrará ejemplos para el tratamiento de eventos por error.

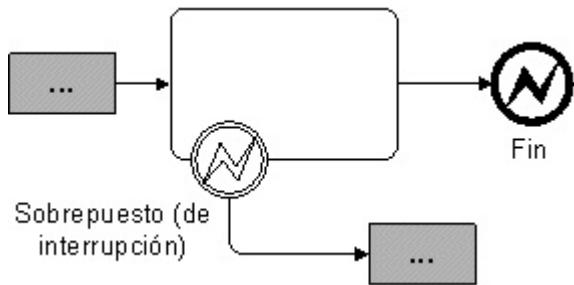


Figura 2.46: Tipos de Eventos de Error

2.6.5. Evento condicional

Podríamos admitir que un proceso puede iniciarse o continuar bajo «ciertas condiciones». Como usted se puede imaginar, una condición puede ser cualquier cosa, pero a diferencia de otros eventos el punto esencial es que la condición debe ocurrir en forma independiente del proceso. Este evento es junto al evento de tiempo el único tipo que sólo existe en forma de evento de captura (ver figura 2.47).

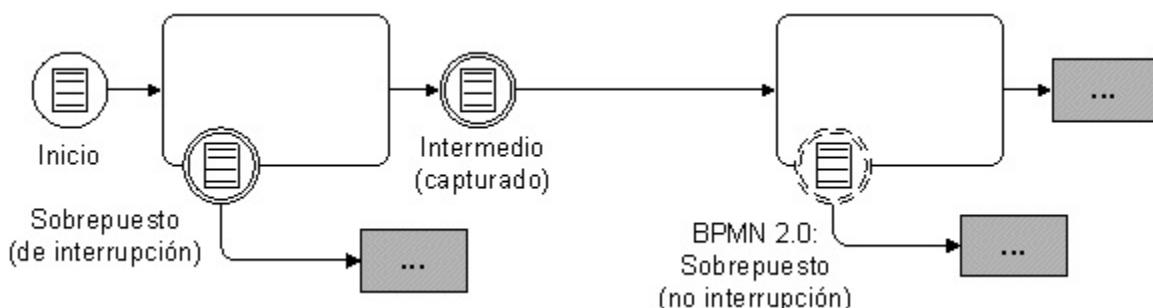


Figura 2.47:

Tipos de Eventos Condicionales

En las figura 2.48 modelamos con eventos de condición el ejemplo de preparar una pizza en el horno. La primera condición es, sólo cuando la temperatura alcance los 180° vamos a poner la pizza en el horno y sólo cuando esté lista (por ejemplo cuando se haya derretido el queso), la vamos a sacar del horno y comer, pero la condición de estar lista es poco precisa, por lo que sería más adecuado de reemplazar el evento de condición «estar lista» por un temporizador como muestra la figura 2.49.

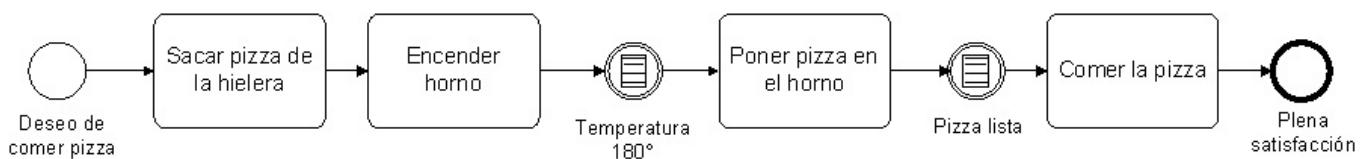


Figura 2.48: Proceso de hornear pizza con eventos condicionales

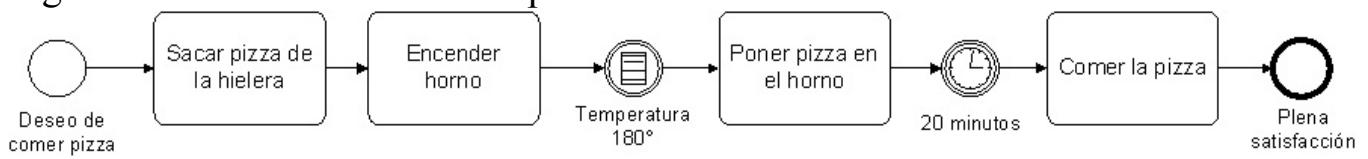


Figura 2.49: Proceso de hornear pizza con evento condicional y temporal

2.6.6. Evento de señal

Los eventos de señal tienen un cierto parecido con los de mensajes, razón por la cual en BPMN las reglas de modelamiento para ambos eventos son iguales (ver figura 2.50). Su símbolo es un triángulo. La única y gran diferencia es que los mensajes tienen un destino definido, por ejemplo un e-mail indica una dirección a quién se dirige y una llamada telefónica indica un número identificador, mientras que una señal es un mensaje con destino indefinido. Un anuncio en el diario, un reclame en la televisión, o un llamado de emergencia por radio son ejemplos de señales.

Cualquiera persona o sistema que capte la señal puede reaccionar si es que quiere.

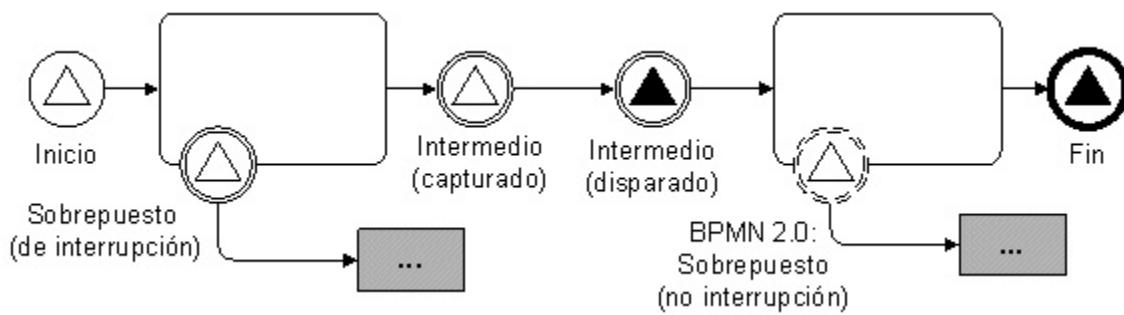


Figura 2.50:

Tipos de Eventos de Señal

En la figura 2.51 modelamos el ejemplo que al ver un comercial en la televisión nos abrió el apetito de probar la pizza tras el anuncio. Entonces llamamos y hacemos el pedido de la pizza (reacción a la señal), pero sólo la comemos cuando tengamos deseo de probarla (evento de condición). Luego evaluamos si nos gustó la pizza en un sitio web de gourmes. Es decir, también los comensales envían una señal (destino indefinido) al evaluar la pizza en un sitio público.

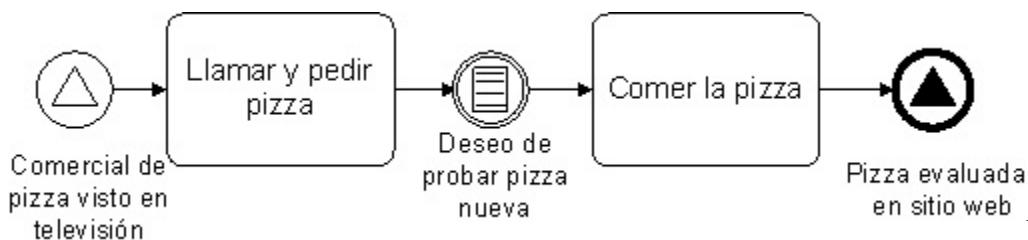


Figura 2.51: Caso

de correcta utilización de Evento de Señal

2.6.7. Evento de término

Para entender como impactan los eventos de término, vamos a analizar las siguientes figuras abstractas. En la sección 2.3.2 analizamos algunos indicadores simples de ciclo, y vimos que este proceso (ver figura 2.52) necesita 55 minutos en procesarse, terminada la actividad 1 pueden ejecutarse en forma simultánea las actividades 2 y 3. La actividad 2 requiere de más tiempo para terminar que la actividad 3, razón que determina el tiempo de ciclo en general. Un token que recorre el proceso es clonado en el AND-Split. El primer token permanece 45 minutos en la actividad 2 y el

segundo token permanece 30 minutos en la actividad 3, entonces el segundo token llega primero al evento de término y es consumido. El primer token necesita aún 15 minutos para que pueda avanzar al evento de término, en donde también es consumido. Como se consumieron todos los tokens, podemos declarar la instancia del proceso como terminada, registrando un tiempo de ciclo de 55 minutos.

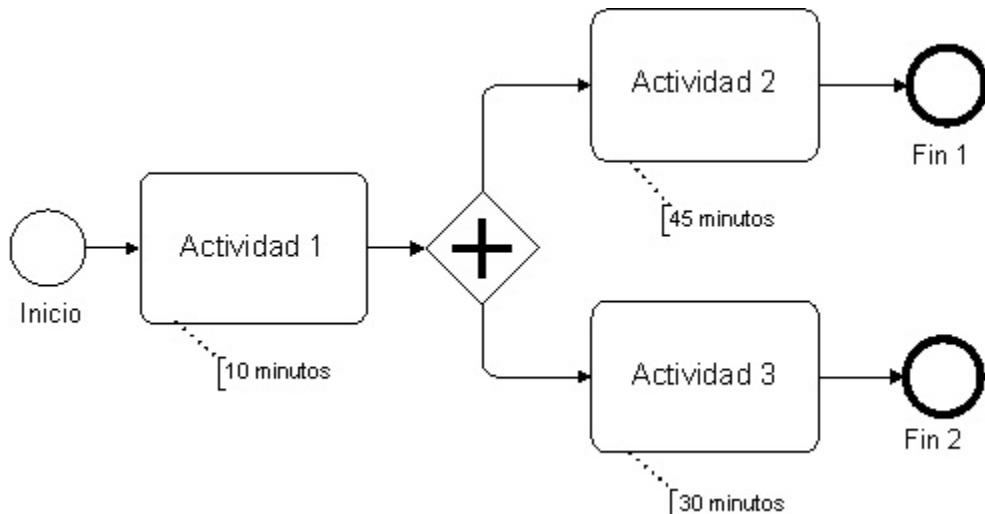


Figura 2.52: Caso de

correcta utilización de evento de término

Ahora ¿cómo modelamos el caso si suponemos que al terminar la actividad 3, puede que la actividad 2 ya no se necesite seguir procesándose? Esta figura se da bastante en la realidad cuando se están procesando dos actividades en paralelo que semánticamente están relacionadas.

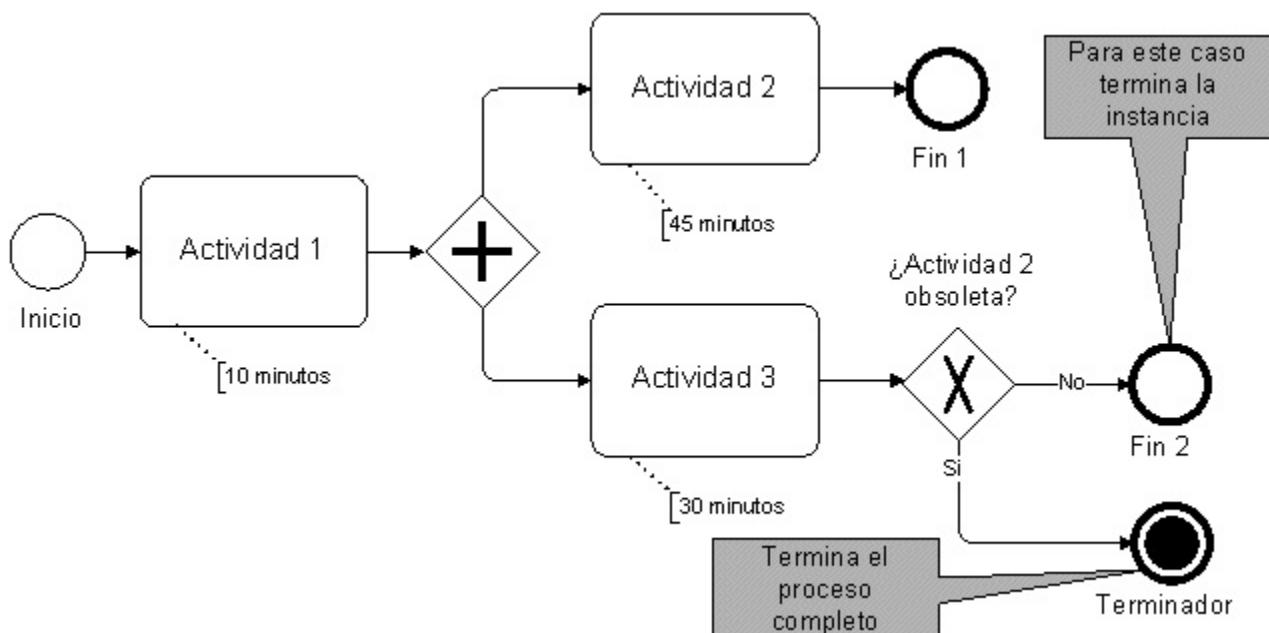


Figura 2.53: Caso de correcta utilización de Evento Terminador

En este caso podríamos modelar el proceso como lo muestra la figura 2.53, que muestra un evento de término justamente creado para estas situaciones. Se le denomina «evento terminador» y su función es buscar todas las marcas activas y

terminarlas (consumirlas).

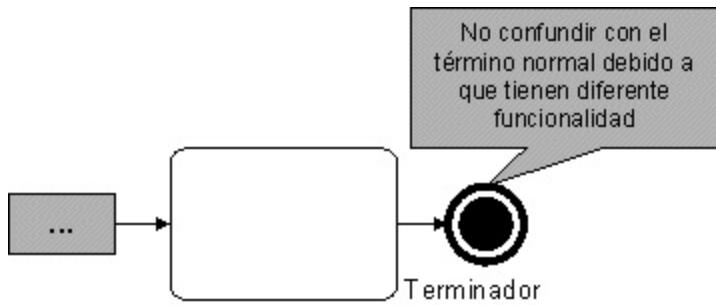


Figura 2.54: El evento terminador

El evento terminador luego de consumir todos los tokens activos se encarga también de finalizar la instancia del proceso. Como consecuencia este evento especial sólo debe usarse como evento final, debido a que termina todos los tokens activos del proceso, independiente de donde se encuentren (ver figura 2.54).

2.6.8. Evento de conexión

El evento de conexión o de vínculo (en inglés: link) es un evento técnico, no tiene ningún significado de negocio. El evento no tiene otra finalidad que poder dividir diagramas muy grandes, sin perder el vínculo de un flujo de secuencia. En la figura 2.55 vemos dos eventos de conexión que corresponden uno con el otro, es decir el de color negro asume la función de impulsar y el de color blanco la función de captar la conexión. Con el fin de no perder el hilo, los analistas de negocio pueden darle un nombre identificador al link, como en nuestro ejemplo la letra A.

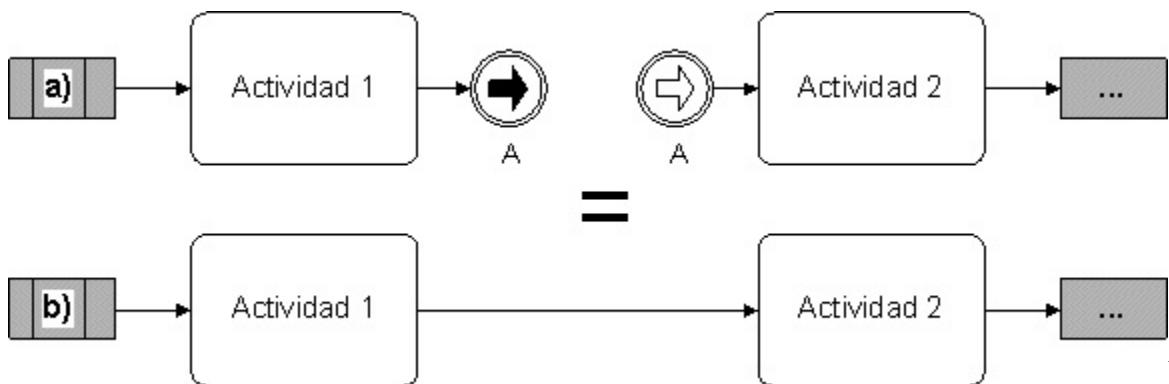


Figura 2.55: Equivalencia entre a) utilización de Eventos de Conexión y b) utilización de flujo de secuencia

Eventos de conexión pueden ser de utilidad cuando:
tenga que dividir un diagrama de procesos en varias páginas,

diagramas de procesos muy intensivos en flujos de secuencia. Para evitar una diagramación tipo «espagetti», se pueden crear subdivisiones técnicas con los eventos de conexión.

Eventos del tipo de conexión sólo se pueden utilizar como eventos intermedios (ver figura 2.56).

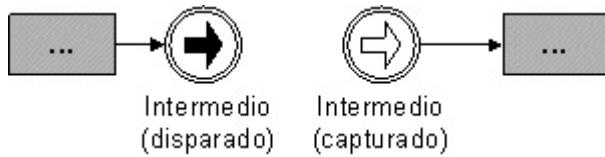


Figura 2.56: Utilización de Eventos de Conexión

2.6.9. Evento de compensación

Compensar en BPMN significa volver al estado inicial de una actividad. En la práctica utilizamos el evento de compensación sólo en el contexto de transacciones que tienen que ser reversadas (ver figura 2.57 y sección 2.85). BPMN también permite «compensar» sin la contemplación de transacciones, pero en la práctica no hemos visto casos en que no estén involucradas las transacciones, razón por la cual las tratamos en conjunto.

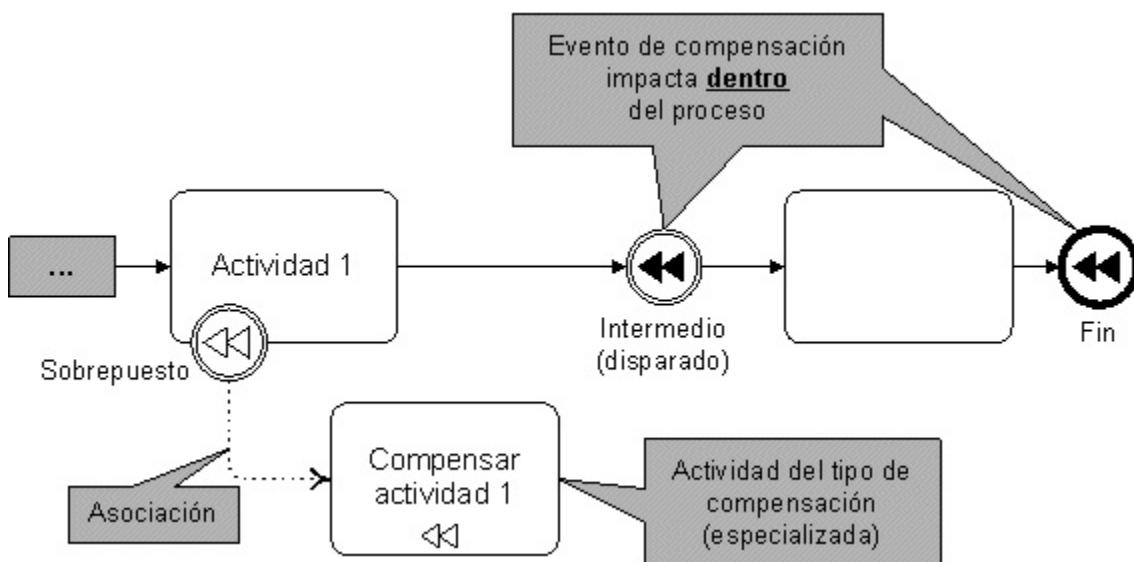


Figura 2.57:

Tipos de eventos de compensación

Típicos ejemplos de compensación podrían ser:

- la reserva de un pasaje
- la reserva de un hotel
- el pago con tarjeta de crédito
- una transferencia electrónica

A continuación podemos apreciar en la figura 2.58 un ejemplo de un proceso de una persona que desea organizar una salida el día viernes después de su jornada laboral. La persona se pone de acuerdo con su pareja después de medio día (13:00 hrs) de organizar una salida al teatro o de invitar a unos amigos a salir. En ambos casos tenemos que comprometernos a reservar las entradas al teatro o de llamar a amigos

para invitarlos. Al atardecer o al llegar a la casa, puede darse la situación de estar muy cansados, cambiar de parecer y quedarnos en casa viendo televisión. Entonces tenemos que cancelar las entradas o llamar a los amigos para manifestarles que no habrá salida.

El evento de compensación permite representar el reverso en forma muy compacta como muestra la figura 2.59. En BPMN no es necesario modelar el proceso de reversa, la notación asume técnicamente todos los pasos necesarios para volver al estado inicial.

Reglas y consideraciones para el empleo de eventos del tipo de compensación:

Eventos de compensación sólo pueden referirse a su propio proceso, es decir la ocurrencia sólo tiene validez dentro de un pool a diferencia de un evento de mensajería que traspasa los límites del proceso.

Eventos de compensación sobrepuertos sólo son válidos si la actividad sobre la cual se encuentra el evento de compensación se haya ejecutada exitosamente. De esta forma se diferencian de todos los otros eventos del tipo interrupción, que interrumpen la actividad durante su ejecución.

Eventos de compensación sobrepuertos se asocian (obligadamente) con una actividad del tipo de compensación con el objeto de «asociación» y no con el objeto de flujo de secuencia. La especificación en BPMN quiere mostrar de esta forma, un tratamiento de excepción y no está inserto en el flujo normal del proceso.

Nuestro Tip en BPMN

En este ejemplo muy sencillo no se aprecia la cantidad de trabajo que hay detrás de una compensación, pero imaginese en procesos transaccionales reales todo lo que habría que modelar para volver al estado inicial de un proceso de negocio. El modelo de compensación en BPMN ahorra todo este modelamiento y logra representar en forma muy compacta el proceso de reversa. Recomendamos usar el evento de compensación sólo para casos transaccionales de negocio y no para reversar errores técnicos.

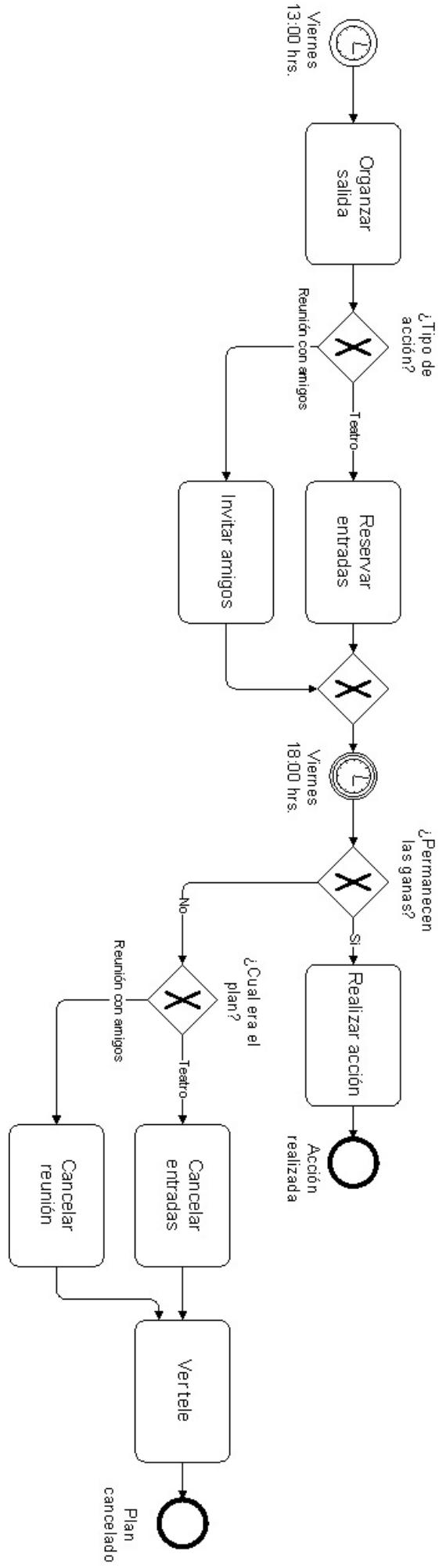


Figura 2.58: Organización de una salida para el día viernes

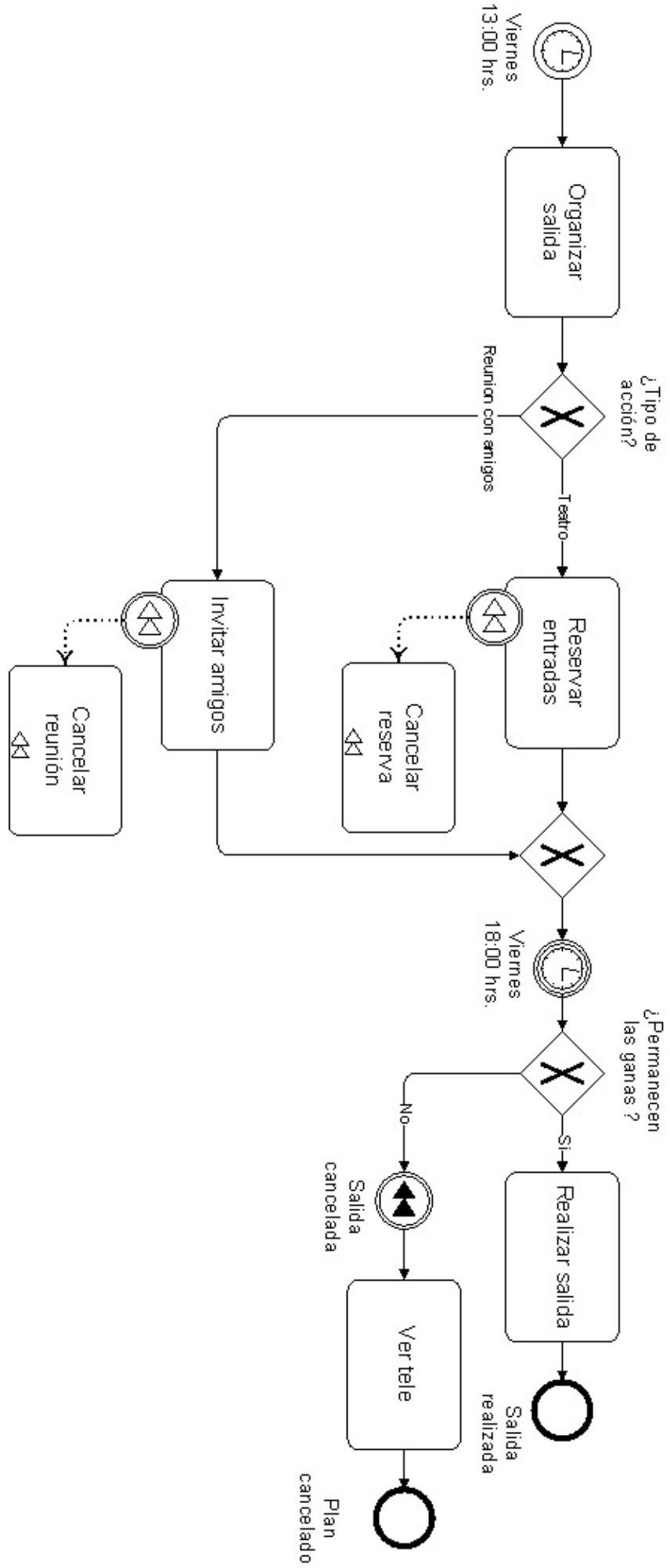


Figura 2.59: Organización de una salida para el día viernes con eventos de compensación

2.6.10. Evento múltiple

Con el evento múltiple podemos incluir la captura de varios eventos alternativos con un símbolo (ver figura 2.60). La semántica es bien simple:

Si se utiliza como evento de captura, inicia o continúa el proceso, con el sólo hecho de ocurrir uno o el primero de los eventos posibles.

Como evento del tipo disparador reacciona como un disparador múltiple, es decir impulsa todos los eventos contenidos.

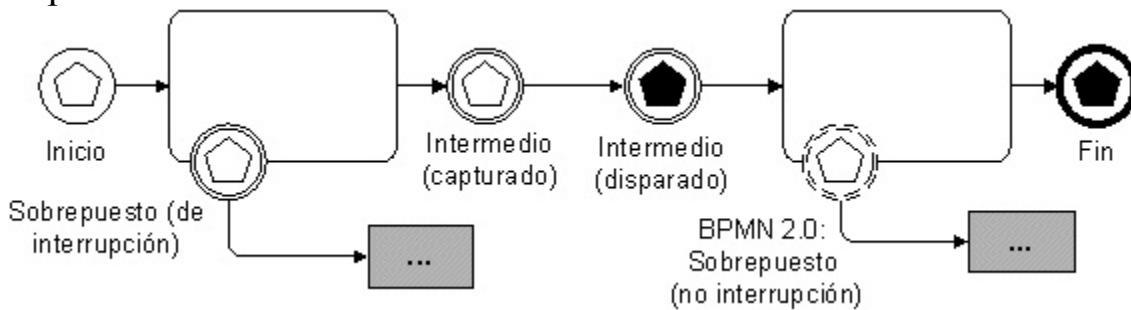


Figura 2.60:

Tipos de Eventos Múltiple

En la figura 2.61 vemos como podría utilizarse este evento en el contexto de nuestro ejemplo de la pizza. En este ejemplo probamos una nueva pizza cada vez que se promocione una nueva o si algún amigo nos recomienda una. Luego de haberla probado, la evaluamos en el sitio web de gourmes y enviamos un reporte a nuestros amigos. El mismo caso lo podríamos modelar sin eventos múltiples como muestra la figura 2.62.

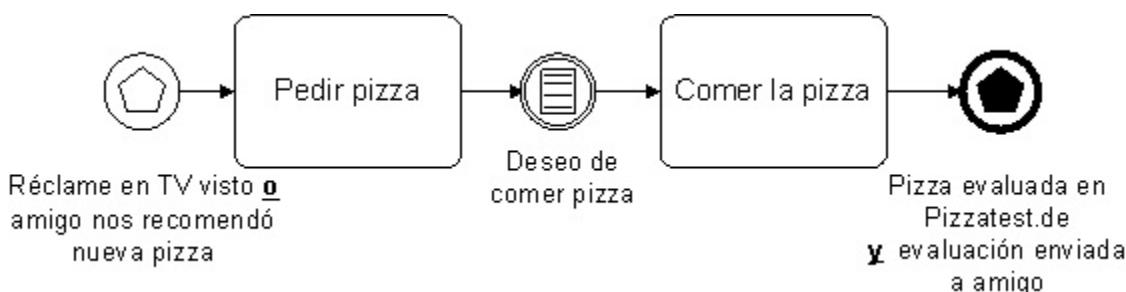


Figura 2.61:

Caso de correcta utilización de Evento Múltiple

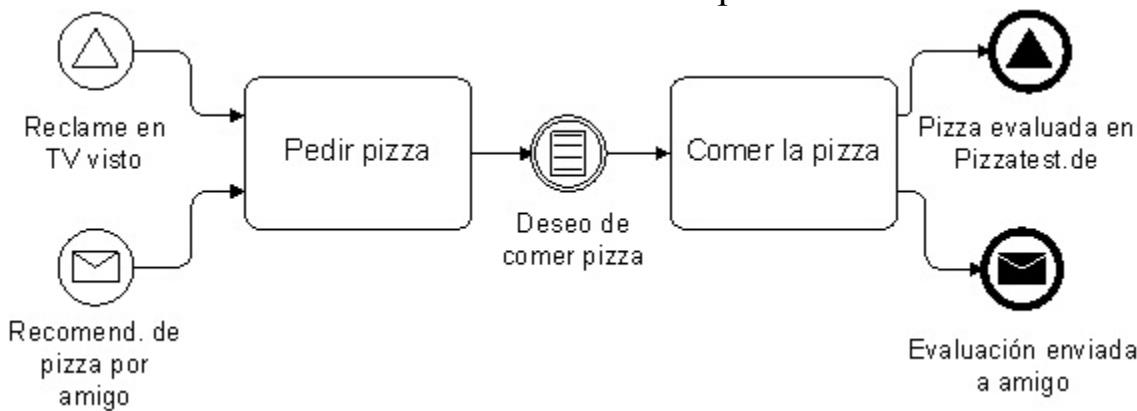


Figura 2.62:

Un modelo alternativo al proceso de la figura 2.61

Nuestro Tip en BPMN

La decisión de emplear este evento múltiple la toma el analista. Nosotros sólo vemos el beneficio de uso de este evento en modelos del nivel descriptivo cuando se trata de compactar o comprimir casos para reducir la complejidad. En el momento de pasar a un diseño técnico se hace necesario separarlo o de programar la condición múltiple (lo cual no es la idea del BPMN). En modelos del nivel operacional no es a nuestro parecer la representación adecuada, por no ser de carácter intuitivo. Los diagramas se verán más amplios, pero al mismo tiempo más comprensivos. En nuestros proyectos no hemos utilizado nunca este evento.

2.6.11. Evento múltiple paralelo

El evento múltiple paralelo se introdujo en la versión BPMN 2.0 como complemento al evento múltiple (ver figura 2.63). Mientras que el evento múltiple se concibe como una semántica del tipo XOR, el evento múltiple paralelo trabaja con una semántica del tipo AND. Es decir tienen que haber ocurrido todos los eventos para que pueda iniciar, continuar o interrumpir una actividad. Como la semántica AND del tipo disparador ya está contenida en el evento múltiple, el evento múltiple paralelo sólo existe en forma de captura (color blanco).



Figura 2.63:

Tipos de eventos de múltiple paralelo

2.6.12. Evento de escalación

El evento del tipo de escalación también es nuevo en BPMN y se utiliza para mejorar el tipo de comunicación entre subprocessos superiores e inferiores (ver figura 2.64). Vamos a describir algunos ejemplos en la sección 2.8 (Subprocesos).

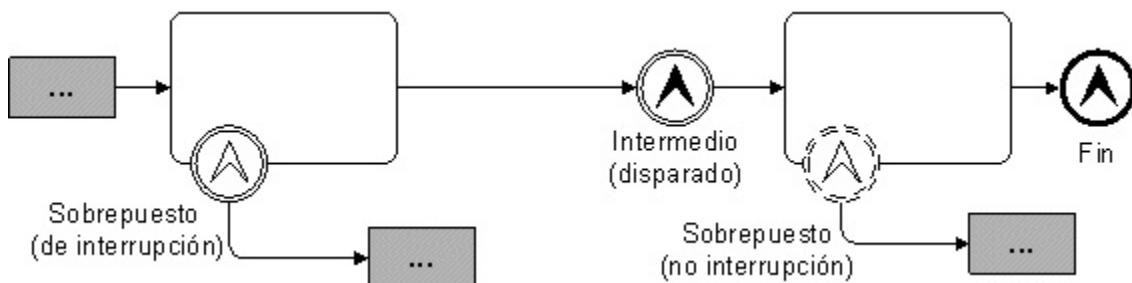


Figura 2.64:

Tipos de eventos de escalación

2.6.13. Evento de cancelación

El evento de cancelación sólo se puede utilizar en el contexto de transacciones, materia que vamos a describir en la sección 2.8.5. Ahí encontrará también ejemplos de uso para este tipo de evento.

2.6.14. Evento de Gateway exclusivo basado en eventos

Con el Gateway exclusivo de datos (XOR-Gateway) conocimos en la sección 2.3.1 una compuerta que nos permite tomar la decisión de rutear el flujo del proceso hacia diferentes alternativas a raíz de la información (datos). En la figura 2.65 mostramos nuevamente el funcionamiento del XOR-Gateway. Primero seleccionamos una receta y dependiendo de la elección el XOR-Gateway nos lleva a las actividades «cocinar pasta» o «freír steak» o «preparar ensalada».

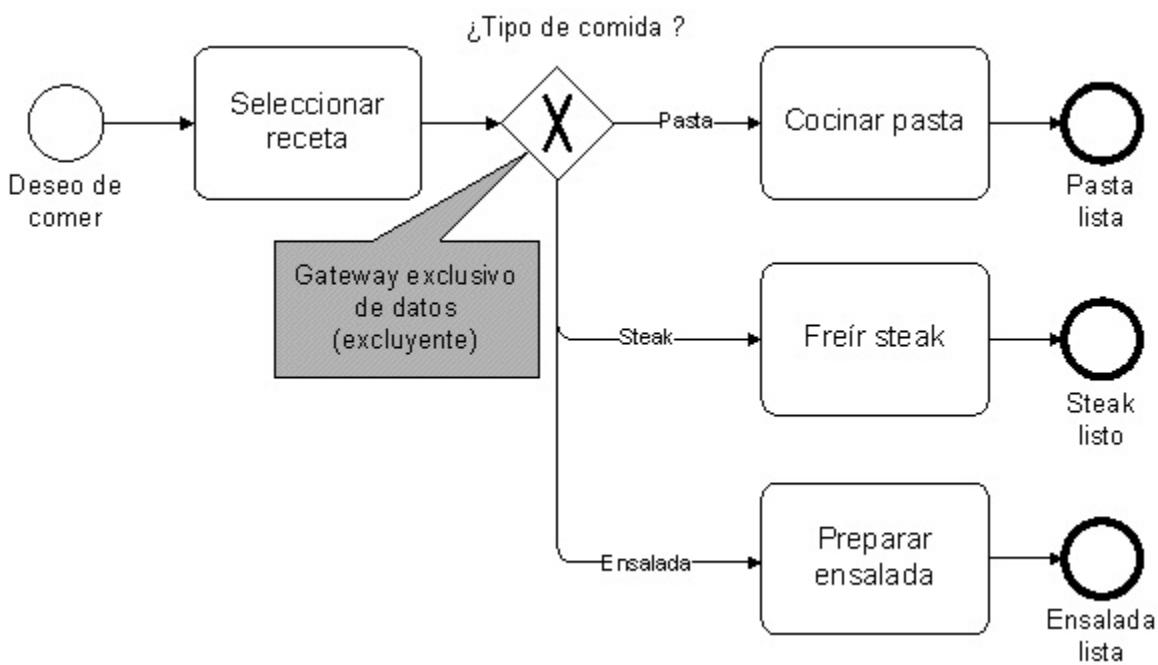


Figura 2.65: Caso con utilización de XOR-Gateway basado en datos

Este tipo de compuerta exclusiva se conoce en todas las notaciones de modelamiento de procesos, pero en BPMN contamos con una posibilidad adicional para diseñar comportamientos especiales, como es el «evento de Gateway exclusivo basado en eventos (abreviado: Event-Gateway)». Este Gateway no reacciona ante datos sino a eventos, específicamente al primer evento que suceda. ¿Qué situaciones se pueden dar para utilizar este Event-Gateway? Para este efecto analicemos nuestro ejemplo en la figura 2.66. Hacemos el pedido de una pizza, esperamos su entrega y luego la podemos comer, pero ¿cómo reacciona el modelo si la pizza no llega? Seguramente vamos a volver a llamar para reclamar el pedido y justamente este es uno de los

casos en donde se emplea el Event-Gateway (ver figura 2.67). A diferencia del XOR-Split, en el Event-Gateway, el token espera al primer evento que ocurra. Apenas esto suceda el token va a recorrer el flujo correspondiente y todos los demás eventos que sucedan más tarde serán ignorados o no serán validos.

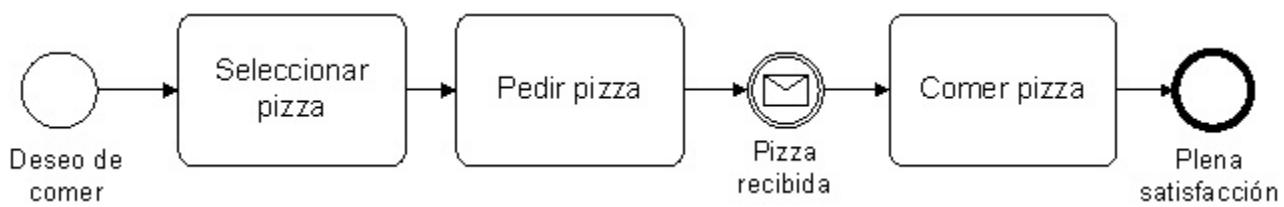


Figura 2.66: Caso con uso deficiente de evento intermedio

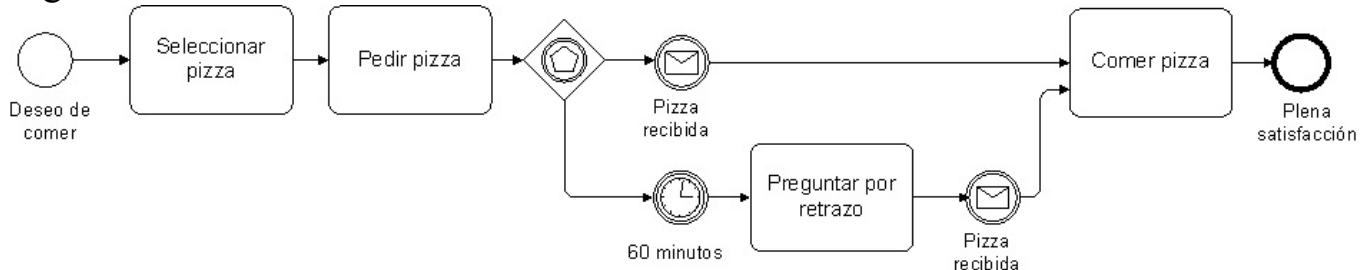


Figura 2.67: Caso de pedido de pizza con Event-Gateway

La figura 2.68 muestra que no todos los tipos de eventos intermedios pueden utilizarse con el Event-Gateway. Como caso excepcional puede combinarse el tipo de actividad de captura, la cual describiremos con mayor detalle en la sección 2.7.

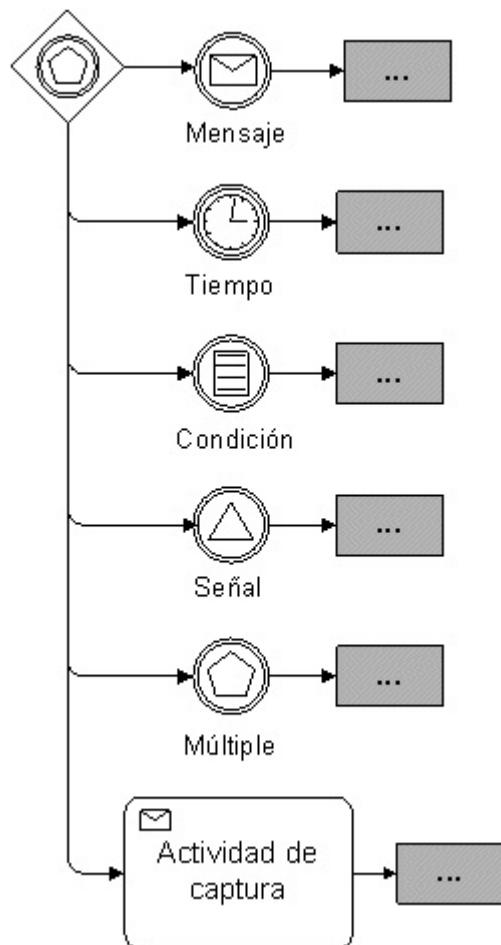


Figura 2.68: Posibles eventos ante los cuales puede

reaccionar el Event-Gateway

El Event-Gateway también se puede utilizar como evento de inicio para un proceso, expresando la notación de esta manera, que diferentes eventos alternativos pueden iniciar un proceso, pero aquí como en muchos otros casos en BPMN contamos con alternativas de modelamiento y el Event-Gateway también lo podemos modelar con eventos de inicio que confluyen en XOR-Join como lo muestra la figura 2.69.

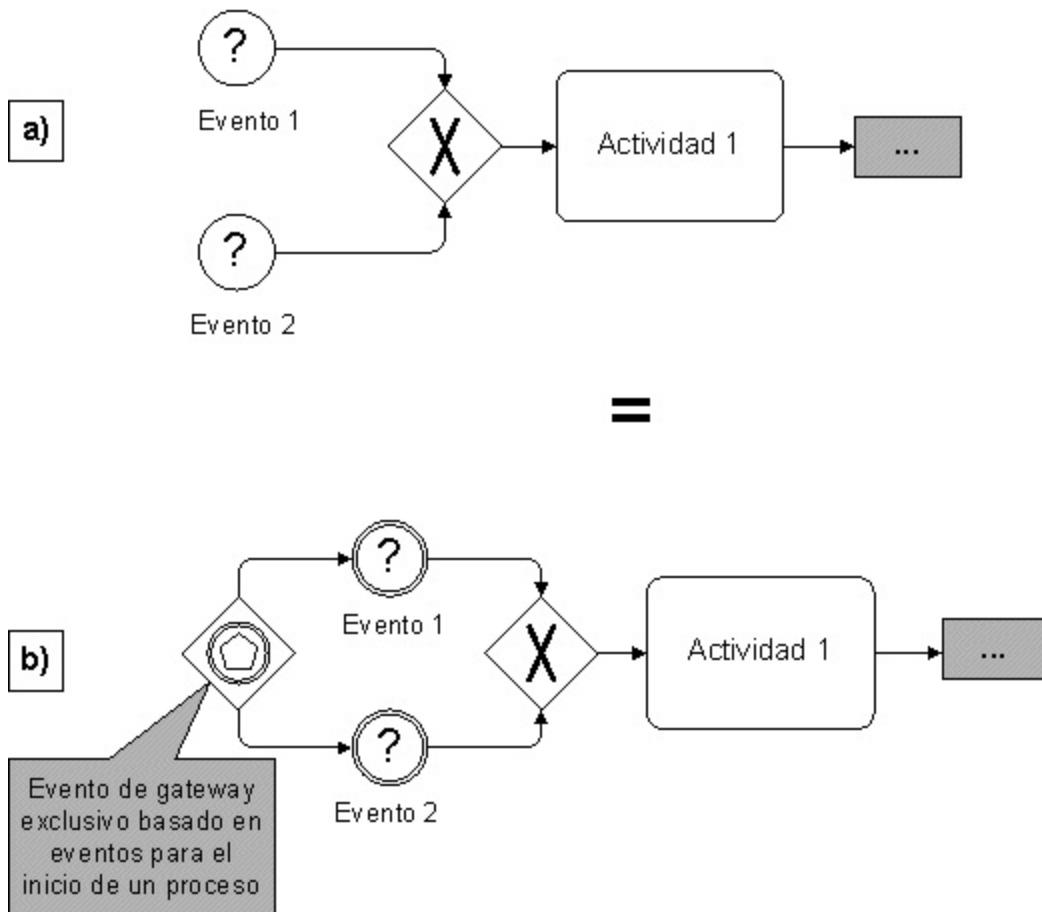


Figura 2.69: Equivalencia entre a) Dos eventos que independientemente pueden gatillar un proceso y b) Evento múltiple como evento de inicio para gatillar un proceso

Nuestro Tip en BPMN

Nosotros encontramos la opción de modelar el caso de diferentes eventos de inicio con el Event-Gateway poco expresivo y preferimos la combinación de eventos en conjunto con el XOR-Join.

2.6.15. Evento de Gateway paralelo basado en eventos

En BPMN 2.0 se incluyó una nueva variante del Event-Gateway, el evento de Gateway paralelo basado en eventos. Con este tipo de objeto se expresa un patrón en el cual todos los eventos indicados tienen que ocurrir para que pueda iniciarse un proceso, es decir expresa aquella correlación que el AND-Join en BPMN no lo

interpreta. En la figura 2.70 vemos el siguiente comportamiento:

Si el evento 1 sucede, se crea una instancia y genera un token, este token espera ahora en el AND-Join.

Cuando el evento 2 ocurra, se identifica la instancia ya creada (correlación) y con este se crea un segundo token.

El segundo token avanza igualmente al AND-Join, donde es fusionado con el primero. Luego sólo un nuevo token (fusionado) sigue el flujo correspondiente.

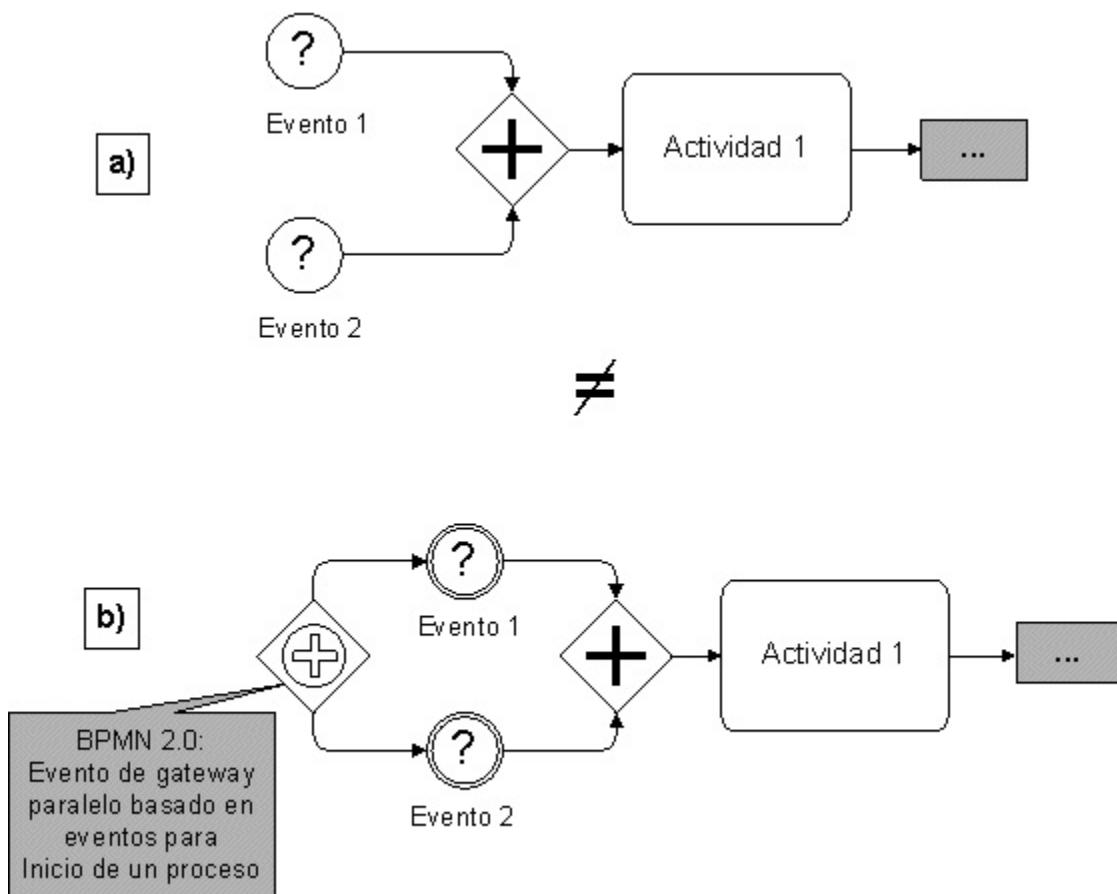


Figura 2.70: Desigualdad entre a) Dos eventos que independientemente pueden gatillar un proceso y b) Evento múltiple paralelo como evento de inicio para gatillar un proceso

En la figura 2.70, en el modelo superior (sin Event-Gateway-Paralelo), no sucedería una sincronización de ambos tokens y se crearían dos instancias. Un token puede quedar esperando por la falta de correlación que suministra la semántica del Event-Gateway-Paralelo. En resumen, no nos agrada mucho esta definición, pero la especificación de BPMN lo describe así.

2.7. Actividades Especiales 2.7.1. Tipos de actividades

En nuestros modelos hemos utilizado hasta el momento sólo actividades del tipo indefinidas. Al igual que en los eventos, BPMN nos ofrece la posibilidad de clasificar las actividades en diferentes tipos. En la práctica no se emplean mucho, porque la tipología fue introducida principalmente para los diseños técnicos, pero tenemos la experiencia que algunos tipos pueden ser de gran utilidad para modelos en la capa de negocio.

Hasta la versión BPMN 1.2 no existía una simbología reservada para diferenciar los tipos de actividades y las que usted haya visto anterior a la versión BPMN 2.0 corresponden a convenciones privadas de los proveedores de herramientas. Esto explica la razón de porque usted a lo mejor se ha encontrado con diferentes simbologías para un mismo tipo de actividad. Esta falencia se superó a partir de BPMN 2.0 y lo que el lector ve en este libro corresponde al nuevo estándar. A continuación describimos los tipos de actividades:

Manual: Una actividad manual es ejecutada por una persona, cuyo control no lo lleva un sistema de workflow o Process Engine. Todos los ejemplos de nuestros procesos relacionados con las pizzas son del tipo manual.

Algunos otros ejemplos:

El guardar un acta en un archivo físico

El aclarar por teléfono una factura mal emitida

La conversación de un ejecutivo con su cliente

Usuario: Una actividad de usuario también es ejecutada por una persona (usuario), pero en este caso el control lo lleva el sistema de workflow o Process Engine, por ejemplo el asignar tareas en su bandeja de entrada. Luego de ejecutarse la tarea por el usuario, el sistema espera que por lo menos le comunique su estado, antes de pueda continuar. El tipo de actividad usuario está inserto en lo que se denomina como «Human Workflow Management».

Algunos ejemplos para Human Workflow Management: Revisar una factura Aprobar una solicitud de vacaciones Administrar una solicitud de soporte

Servicio: Una actividad tipo servicio es una actividad automática que es ejecutada completamente por algún software. BPMN parte normalmente de la base, que se trata de un servicio web, pero no es mandatorio. De todas formas se trata de una componente de integración de aplicaciones, con lo cual tenemos por esta vía la entrada al puente con las arquitecturas orientadas al servicio (SOA).

Ejemplo de servicios de integración: Solicitud de clasificación de riesgo crediticio a un sistema interbancario Verificación de stock de bodega para una orden de compra Disponibilidad de asiento para una reserva de pasajes

Enviar y Recibir: La recepción de un mensaje en BPMN puede modelarse como una actividad aunque exista un tipo de evento para estos fines. Si se desea que un proceso sea instanciado por una actividad de tipo recepción y de esta forma reemplazar un evento del tipo mensaje al inicio de un proceso, se debe utilizar el símbolo de tipo de carta con un círculo (ver figura 2.71). El mismo principio tiene validez para las actividades del tipo envío. Este tipo de actividades son solamente técnicas y se usan para invocar interfaces asincrónicas de servicios web en colas de mensajería (en inglés: message queues), etc., por lo que no se recomienda de usarlas en modelos de negocio.

Script: Un script es un pequeño programa que puede interpretar y ejecutar directamente el sistema de workflow. El script tiene que estar escrito en el lenguaje que pueda interpretar el entorno de implementación.

Regla de Negocio: Este tipo de actividad también es nuevo a partir de BPMN 2.0 y se interpreta de tal forma que sólo está destinada para ejecutar una regla de negocio, ya sea invocándola a un sistema independiente (BRMS: Business Rule Management System) o ejecutando un motor de reglas que viene incluido en el Process Engine.

Tipos propietarios de actividades: La mayoría de la gente no sabe que BPMN también permite definir sus propios tipos de actividades, suponiendo que la herramienta que utiliza la organización lo permita. Hasta el momento no nos hemos encontrado con modelos que hayan hecho uso de esta posibilidad, pero si para un negocio es importante destacar algún tipo de actividad, ¿por qué no hacer uso de ella? Podríamos pensar en:

Firma digital

Tipo de llamada en un callcenter

Aprobaciones y desaprobaciones etc..



tipos de actividades en BPMN 2.0

Figura 2.71: Simbología para

2.7.2. Propiedades de actividades

Además de la clasificación de las actividades, podemos marcarlas con ciertas propiedades (marcas), tales como repetitivas (loop), múltiples (más de una instancia), o compensación. Estas marcas pueden combinarse con los tipos de actividades, convirtiéndose de cierta forma en actividades complejas. A continuación se describen las distintas propiedades de las actividades:

Loop: Una actividad con la propiedad de loop (bucle) se va a repetir tantas veces hasta que se cumpla, o no se cumpla la condición especificada. Por ejemplo, vamos a darle a escoger diversos menús a nuestros amigos, hasta que todos se hayan puesto de acuerdo sobre cual de ellos se va a preparar (ver figura 2.72).

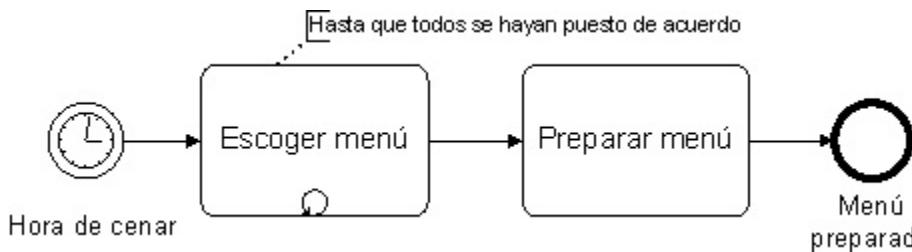


Figura 2.72: Caso con uso de actividad con propiedad de loop

¿Es necesario usar este símbolo para representar este caso en BPMN? Por supuesto que no, también podemos modelar la condición repetitiva con ayuda de Gateways, en combinación con Gateways o sin ellos como lo muestra la figura 2.73.

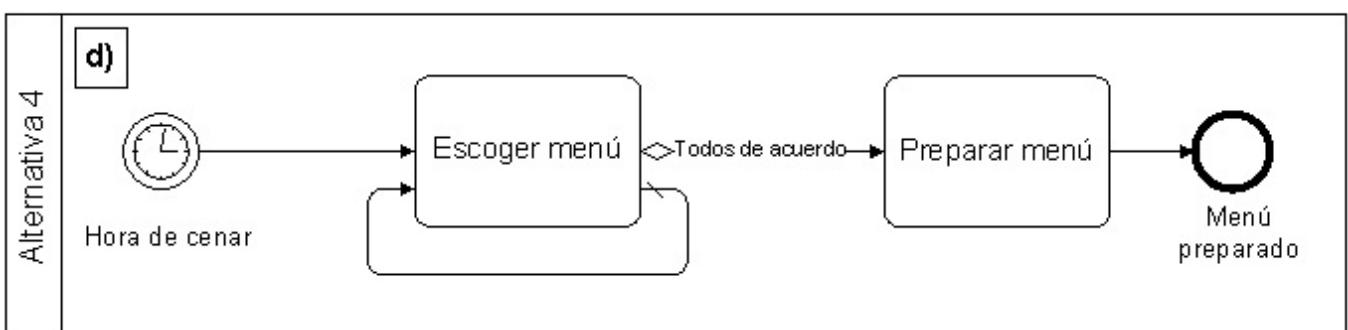
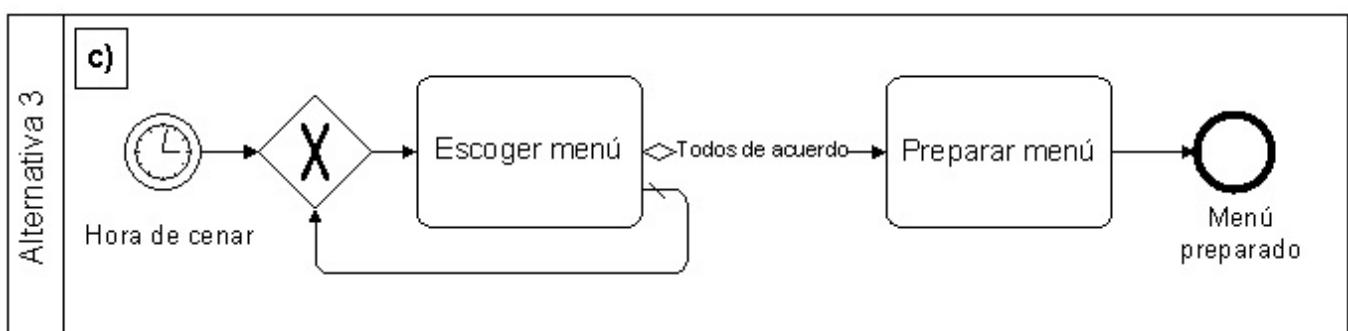
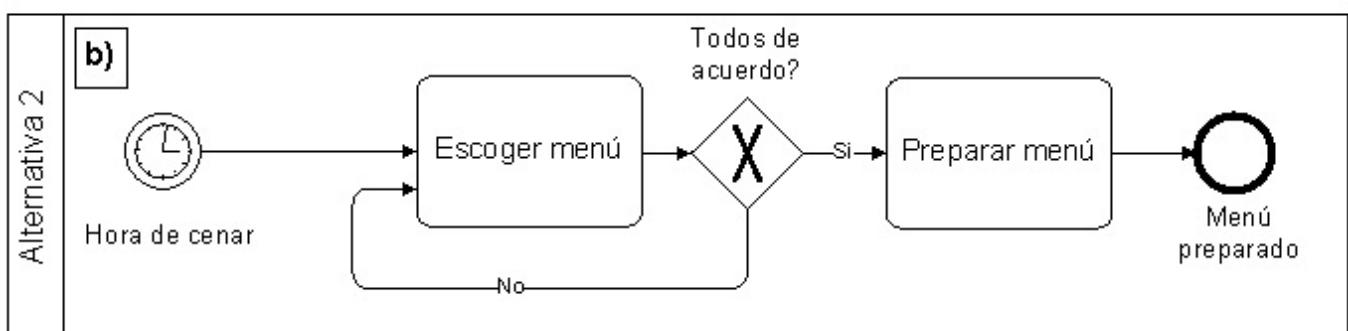
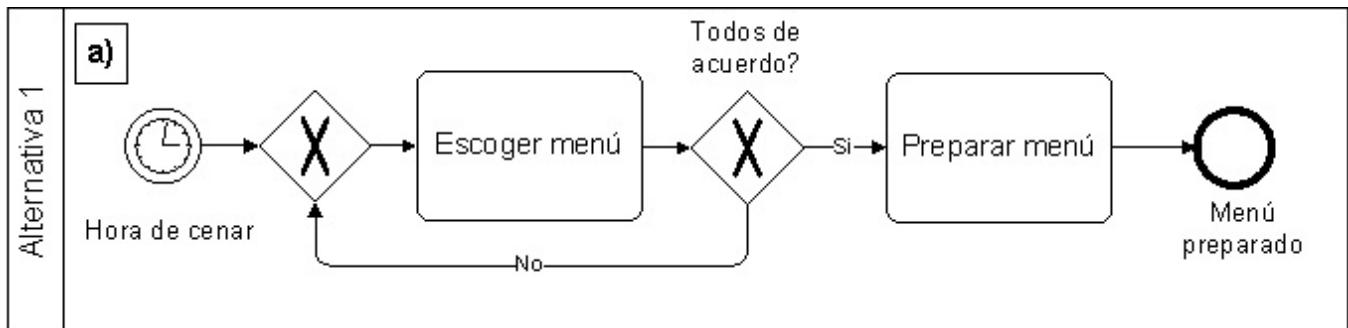


Figura 2.73: Cuatro casos que representan el mismo proceso de la figura 2.72. a) Alternativa 1 con utilización de XOR-Join y XOR-Split b) Alternativa 2 con utilización de XOR-Split c) Alternativa 3 con utilización de XOR-Join y flujos condicionales d) Alternativa 4 con utilización de flujos condicionales

¿Cuál de las siguientes alternativas de la figura 2.73 prefiere usted? Todas son correctas, no infringen reglas sintácticas y corresponden en su semántica con el proceso modelado en la figura 2.72. Cualquiera de las alternativas que su organización escoja como convención para modelar es válida, sólo recomendamos no mezclar las diferentes técnicas que representan lo mismo.

En nuestro ejemplo ejecutamos primero la actividad y luego preguntamos si se cumple la condición o si tenemos que volver a repetir la actividad. Los programadores conocen este principio como un loop del tipo «do-while», pero también se puede construir este caso al revés, es decir primero preguntar por la condición y luego ejecutar la actividad, con lo cual estaríamos aplicando el principio de «while do» en la programación. Normalmente se usa más el primer principio y el segundo cuando se pueda dar la constelación que no sea necesario ejecutar la actividad por primera vez.

El uso de la marca loop puede compactar bastante un modelo si no se muestran mayormente propiedades de negocio en el bucle y en el diseño técnico no debiera existir ningún problema en implementar esta propiedad.

Actividad múltiple: A diferencia de la actividad con propiedad de loop, la cual repite la ejecución en forma secuencial, la actividad con propiedad de múltiple ejecuta en forma simultánea la actividad tantas veces como instancias existan. En nuestro ejemplo de la pizza se daría la siguiente situación para transparentar la diferencia de ambas propiedades. Los cohabitantes de una residencia estudiantil quieren pedir una pizza familiar y en el primer caso tienen sólo una carta de menú, de forma que cada uno hace su elección y luego se la pasa al otro y esto tantas veces, hasta que se pongan de acuerdo (caso loop). Otra alternativa sería que pongan la carta al medio de la mesa de comer y todos se agachen hacia adelante y negocien la elección de la pizza (caso múltiple). Un caso múltiple también podría ser que cada uno tenga una carta de menú, lo cual podría ahorrar ciertas incomodidades del caso anterior. El proceso con propiedad múltiple lo encuentra en la figura 2.74 en la cual la actividad lleva un símbolo de tres líneas verticales. Una actividad múltiple se puede instanciar también en forma secuencial, pero siempre será más eficiente poder hacerlo en forma paralela.

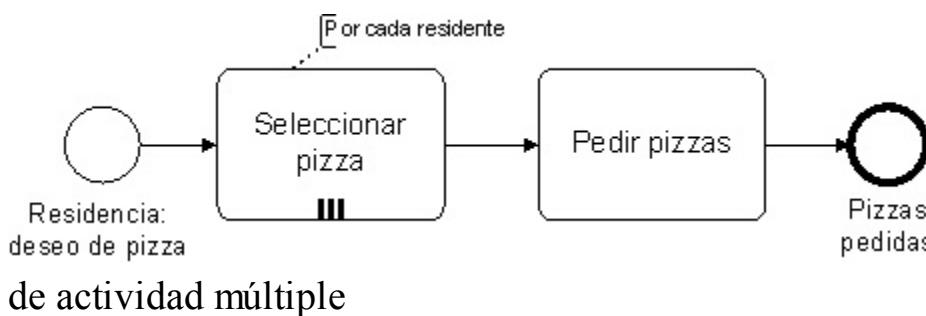


Figura 2.74: Caso con uso de actividad múltiple

¿Encuentra el ejemplo poco común? Cómo se revisa por ejemplo en su organización una factura que contiene múltiples servicios, supongamos una cuenta única telefónica para una empresa. Se envían los comprobantes a cada departamento en forma secuencial, o se envía a cada uno de ellos en forma simultánea el registro de llamadas para que los revisen. Aunque usted no lo crea, la automatización del control de facturas o documentos y en general los procesos que se puedan paralelizar, son principalmente los objetivos que se persiguen, hasta ahora en

proyectos de optimización de procesos.

Actividad de compensación: El beneficio que se tiene en una notación de modelado de procesos al contar con actividad del tipo de compensación, se explica en la sección 2.6.9 junto a un ejemplo. Esta propiedad se utiliza exclusivamente en el contexto del evento intermedio de compensación y se relaciona con el evento a través del flujo de asociación y nunca a través del flujo de secuencia.

Cabe destacar aún la posible combinación con la propiedades del tipo loop o de instancia múltiple como muestra la figura 2.75. En este caso, ambas marcas se posicionan una al lado de la otra, como también es posible de combinarlas con ciertos tipos de actividades, como por ejemplo la compensación del tipo manual o automática (ver figura 2.75).

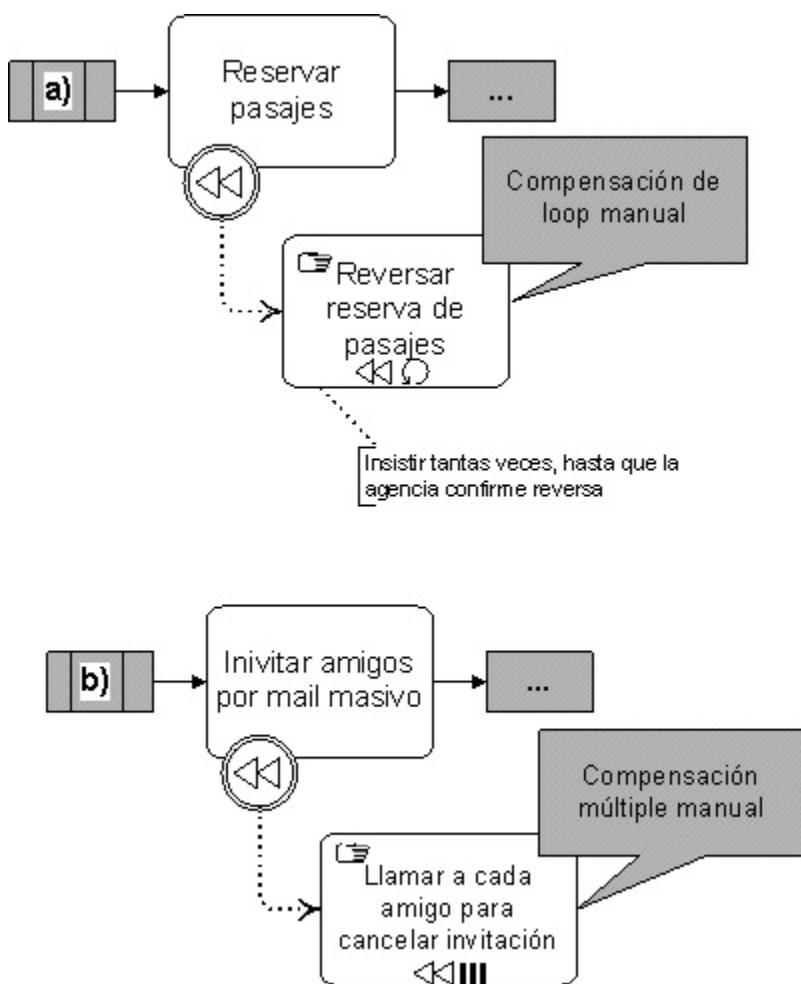


Figura 2.75: Casos con uso de Actividad de compensación a) Compensación de loop manual b) Compensación múltiple manual

2.7.3. Actividades globales

A partir de la versión BPMN 2.0 podemos definir actividades globales, las cuales se diferencian de las actividades normales, en que las primeras pueden reutilizarse. Las actividades globales no se diferencian gráficamente de las normales. Sólo las

reconocemos al ver la existencia de actividades invocables que llevan el mismo nombre que las globales, pero tienen una línea de contorno más gruesa, es decir en negrita como lo muestra la figura 2.76.

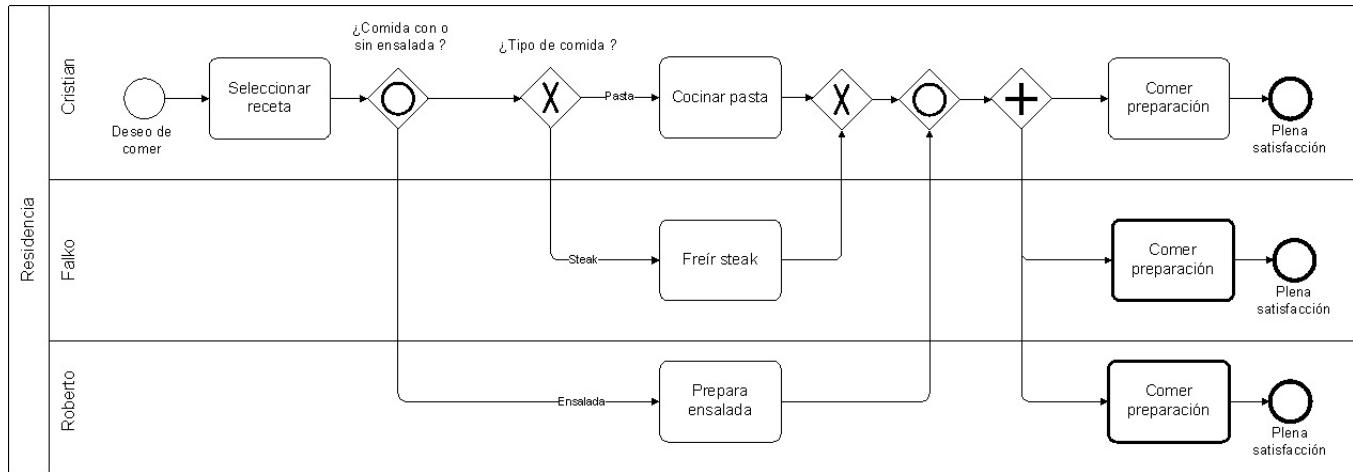


Figura 2.76: Actividades globales segun BPMN 2.0

2.8. Subprocesos 2.8.1. Reducción de complejidad

Los ejemplos que tratamos en este libro los mantenemos sencillos con una doble finalidad, primero por aspectos pedagógicos y segundo para que alcancen en una hoja de libro. Si nos transportamos a la realidad, no nos podemos permitir este lujo. Usted tiene que cumplir con el desafío de mantener por un lado los mapas de procesos abstraídos de la complejidad con el fin de no perder la visibilidad y por el otro lado tiene que describir toda la lógica operacional en detalle para que puedan ser implementados y para identificar potenciales de mejora. Justamente el juego entre descomposición top-down y agregación bottom-up distingue buenos modelos de procesos de diagramas de flujo banales y también buenas herramientas de BPM de herramientas triviales.

En BPMN tenemos para estos fines el objeto de flujo llamado subprocesso. Un subprocesso describe en su interior la lógica en detalle, pero en el diagrama del proceso superior no toma más lugar que una propia actividad. Ambos elementos, la actividad y el subprocesso, pertenecen a la clase de las actividades y se representan en forma de rectángulo con esquinas redondeadas. La única característica que los diferencia es un signo más (+) en la actividad del tipo subprocesso, que indica la existencia de una lógica dentro de este (ver figura 2.77).

Tenemos que admitir que un buen balanceo de diferentes niveles de abstracción, depende fuertemente de como una herramienta de BPMN apoye las siguientes funcionalidades:

1. Representación del subprocesso en un diagrama propio: El signo (+) debe estar

relacionado con un nuevo diagrama de proceso. Por ejemplo si su herramienta

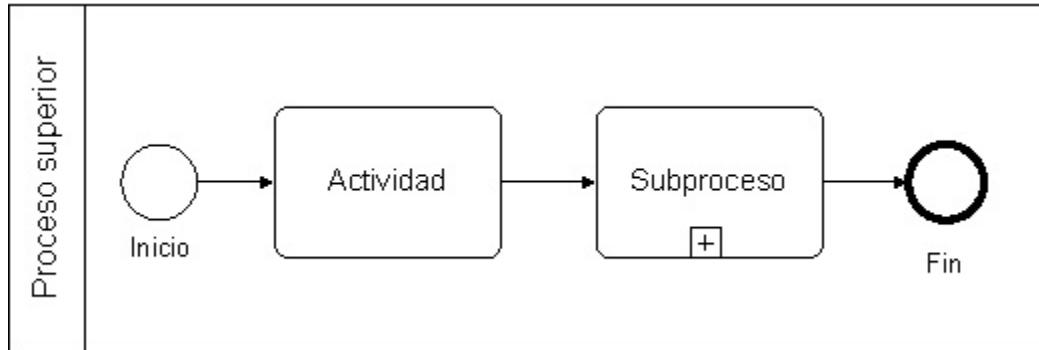


Figura 2.77: Una

actividad y un subprocesso
de BPMN trabaja full web, entonces podría abrir una ventana browser por cada
diagrama (ver figura 2.78 en la página siguiente).

2. Expandir un subprocesso en un mismo diagrama del proceso superior: La especificación de BPMN define un subprocesso con el signo más (+), como un subprocesso cerrado. El signo más indica que si uno hace clic sobre él, debería de expandirse o abrirse. La figura 2.79 muestra en forma ejemplar como BPMN indica que debería ser, pero por razones técnicas que vamos a ver más adelante, los proveedores de herramientas prácticamente no apoyan esta tecnología.

A pesar de las buenas intenciones que pueden haber detrás de este objetivo, de expandir y comprimir los detalles de los modelos de los subprocessos, en la práctica esto no es viable para subprocessos complejos. Al expandir un subprocesso tienen que reposicionarse todos los demás objetos del diagrama para hacer lugar al subprocesso expandido. Como consecuencia, se debe disminuir el tamaño de los objetos tanto como tenga lugar el o los subprocessos expandidos, lo que tampoco puede ser el objetivo de los creadores de este concepto. Lo que si podemos hacer, y en la práctica se usa, es modelar subprocessos simples en forma expandida en el diagrama principal, de forma que no perturbe el orden del diagrama de flujo en general. Esto se hace en subprocessos que tienen eventos de excepción sobrepuertos (ver sección 2.8.3).

En ambos casos termina el flujo de secuencia del proceso superior en la linea izquierda del subprocesso y continúa por fuera por el lado derecho, es decir el flujo de secuencia no debe ingresar al subprocesso, lo que a algún principiante le cuesta entender cuando trabajamos con subprocessos expandidos. Si aplicamos el concepto de token, tendríamos el siguiente comportamiento:

El proceso superior se inicia y nace un nuevo token.

El token pasa por la actividad y llega al subprocesso. Esto conlleva a que el proceso superior cree una instancia del subprocesso.

Dentro del subprocesso se crea un nuevo token que sigue la lógica del flujo del subprocesso desde el evento de inicio hasta el evento que termina el subprocesso. El

token del evento superior espera el arribo del token del subprocesso.

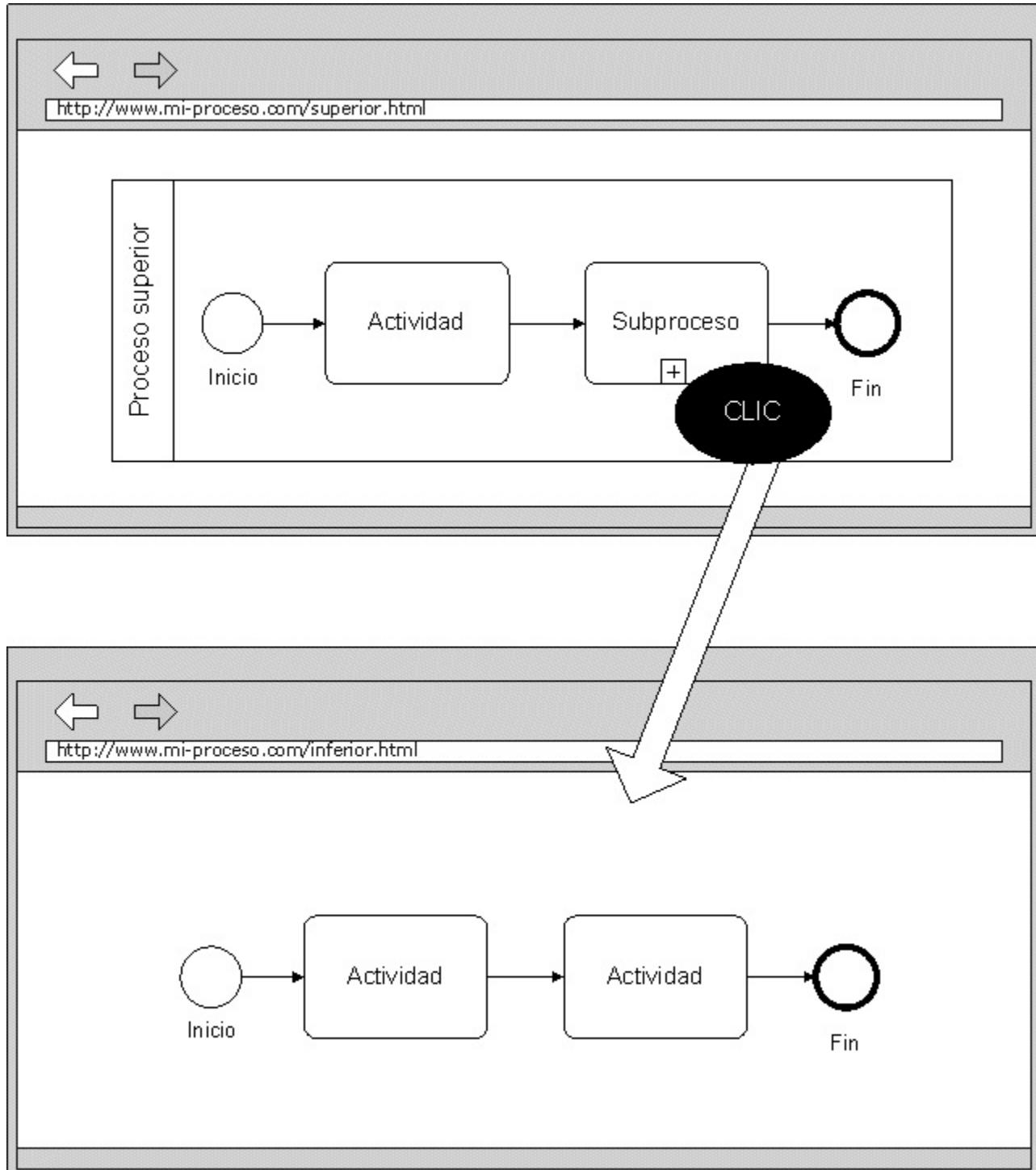


Figura 2.78: Caso con representación de subprocesso en un diagrama propio por separado

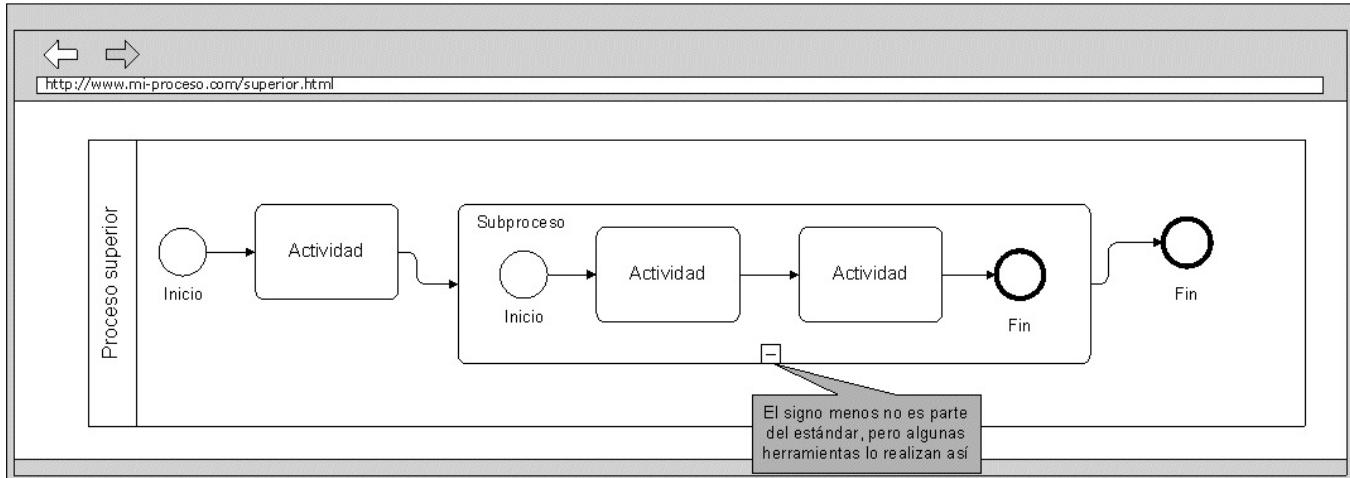


Figura 2.79: Caso con representación de subprocesso en el mismo diagrama principal del proceso

Cuando el subprocesso gatilla su evento de finalización es consumido, el subprocesso ha terminado y el token del proceso superior continúa hasta llegar a su propio fin.

Lógicamente que la cantidad de niveles que puedan tener procesos superiores e inferiores (todas las jerarquías representadas a través de subprocessos) no se restringe sólo a dos niveles. El proceso superior puede a su vez ser subprocesso, es decir inferior a otro superior, etc.. BPMN no hace ninguna limitación en cuanto a la cantidad de niveles de abstracción.

BPMN permite también posicionar los eventos de inicio y de término de los subprocessos directamente en los límites del subprocesso, como lo muestra la figura 2.80. Lógicamente que esto sólo funciona si el subprocesso se encuentra expandido en el proceso superior. En la práctica no conocemos herramientas que apoyen esta funcionalidad y nosotros tampoco vemos algún beneficio en este tipo de diagramación, por lo que tampoco lo recomendamos.

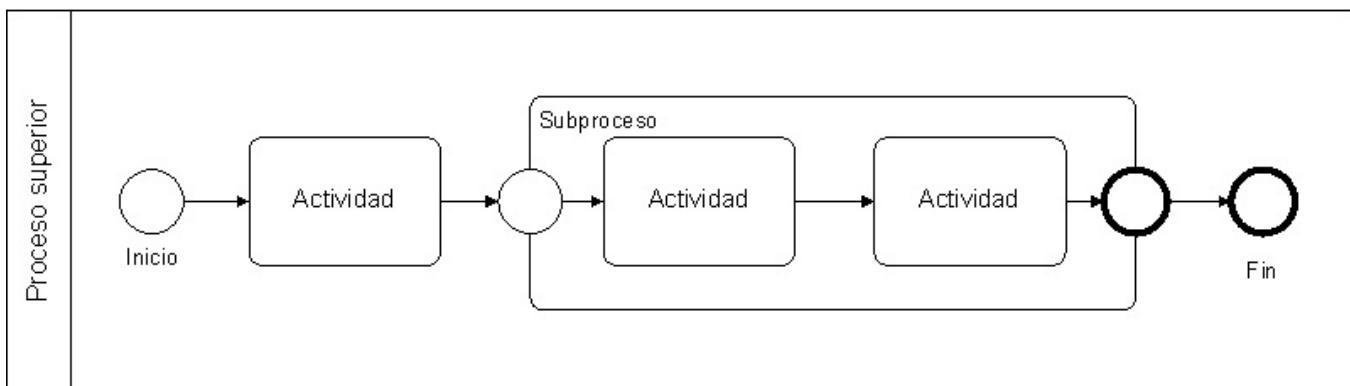


Figura 2.80: Caso con posicionamiento de eventos de inicio y término en los límites del subprocesso

A lo mejor usted se acuerda que en la sección 2.2 explicamos que en un principio BPMN permite modelar sin eventos de inicio y de término. Uno podría aprovechar esta posibilidad para hacer más compactos algunos diagramas y convertirlos en

subprocesos como muestra la figura 2.81. Un subprocesso también puede reemplazar una construcción de AND-Gateways como muestra la figura, pero nosotros no recomendamos esta práctica por los siguientes dos motivos:

1. Aumenta el peligro que lectores que no sean expertos en BPMN se confundan.
2. Uno podría confundir rápidamente esta representación con los subprocessos del tipo ad-hoc, como lo vamos a ver más adelante. Por otro lado, los subprocessos del tipo ad-hoc se emplean bastante en la práctica.

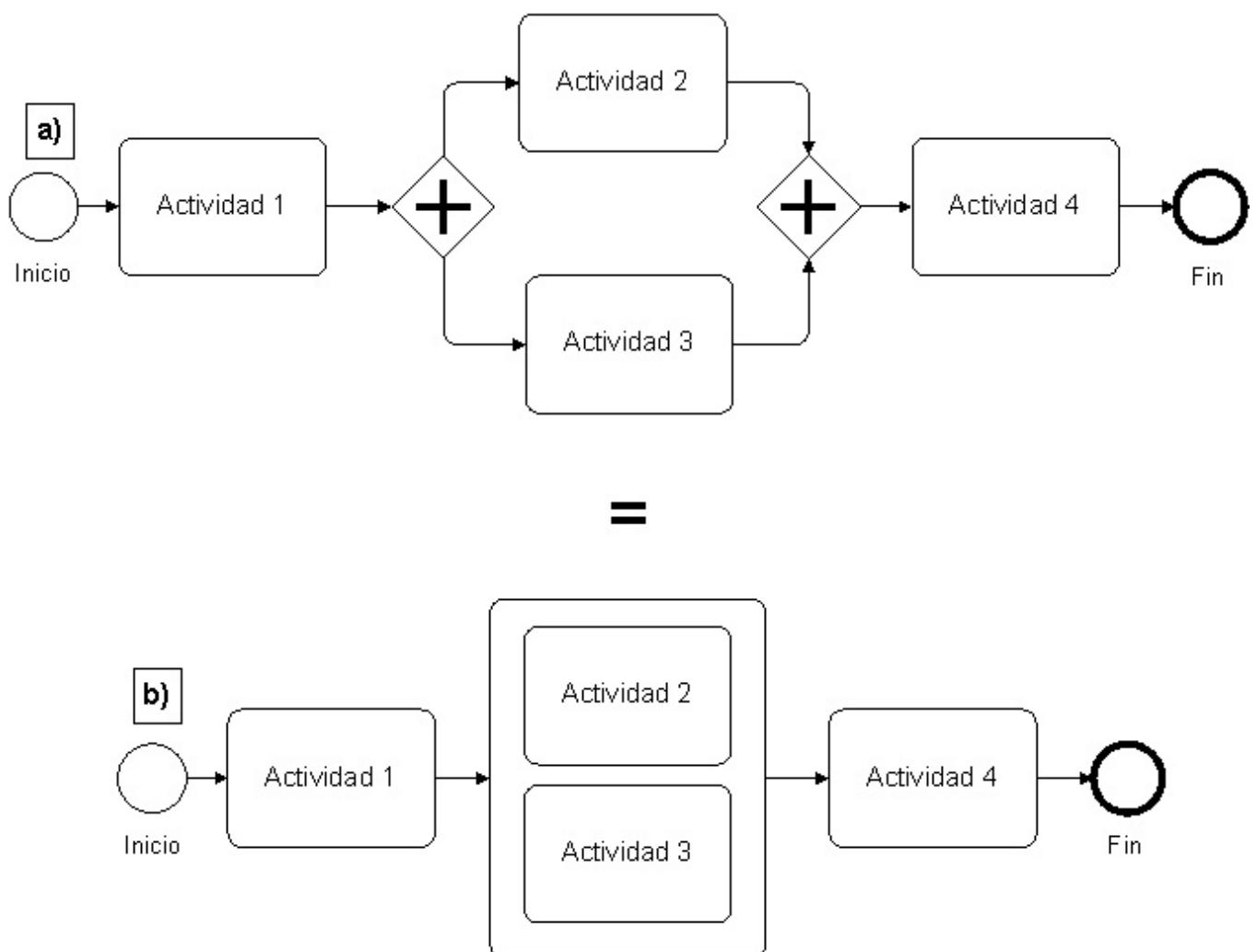


Figura 2.81: Equivalencia entre a) Caso con utilización de AND-Gateway y b) Caso con uso de subprocesso

Como usted puede deducir hasta el momento, los subprocessos en BPMN no sólo se utilizan como medio de abstracción metodológica, sino también como medios y técnicas de diseño (estilo) de los diagramas. Las siguientes secciones van a reforzar lo dicho.

2.8.2. Modularizar y reutilizar

En la versión anterior de BPMN 1.2 se diferenciaba entre subprocessos «incrustados» y subprocessos «reutilizables». En un principio permanece aún esta diferenciación, pero con la salvedad que en BPMN 2.0 todos los subprocessos tienen la características de ser «incrustados» y sólo pueden tomar la propiedad como «reutilizable» si se declara como subprocesso «global» y es referenciado como una actividad «invocable». A continuación nos referimos entonces a subprocessos incrustados y globales.

Un subprocesso incrustado sólo puede existir en un proceso superior, dicho de otra forma le pertenece al proceso superior. Un subprocesso incrustado tampoco puede poseer pools o lanes propios, sino que sólo puede estar relacionado con el pool o lane en donde está asociado al proceso superior. Otra regla estricta es que un subprocesso incrustado sólo puede iniciarse con un evento indefinido, otros eventos de inicio como de mensajería o de tiempo no están permitidos. Dicho de otra forma, un subprocesso incrustado, no es otra cosa que una especie de dominio delimitado (en inglés: «scope») dentro del dominio superior, que obedece a dos objetivos:

1. Abstracción de complejidad.
2. Una agrupación de actividades del proceso superior, al cual se pueden relacionar marcadores o eventos en general. Estos aspectos los vamos a tratar más adelante.

En cambio subprocessos globales pueden reutilizarse en varios procesos superiores. En los procesos industriales existen muchos subprocessos que se utilizan en forma global, como por ejemplo la adquisición de un artículo debido al pedido de un cliente o la reposición de stock de bodega. Otro ejemplo de subprocesso global es la facturación representada en la figura 2.82 para la venta de productos o servicios de reparación. En este ejemplo también reconocemos que las actividades invocables están marcadas en su contorno con negrita.

En diferencia a los subprocessos incrustados, los de tipo global no están ligados al proceso superior, por lo tanto si pueden poseer sus propios pools y lanes. Los subprocessos globales se pueden comparar con un servicio externo (que también lo puede ser), es decir como una especie de «Shared-Service-Center».

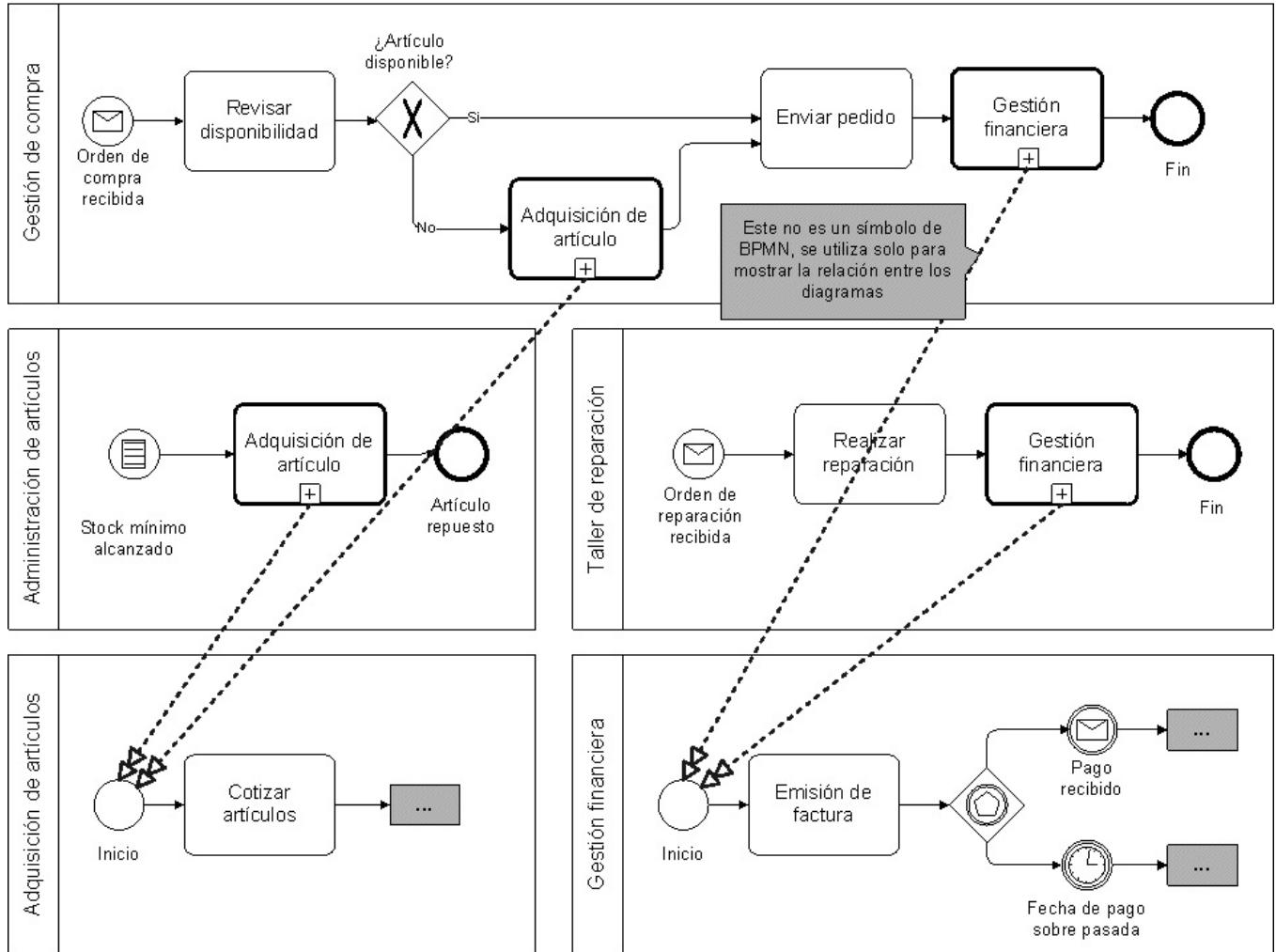


Figura 2.82: Ejemplos para la reutilización de subprocessos

El bajo acoplamiento de los subprocessos globales también se manifiesta en la forma de traspaso de datos a los procesos superiores. BPMN parte de la base que los procesos incrustados tienen acceso a toda la información requerida del proceso superior, en cambio para los subprocessos globales es necesario declarar la forma de traspaso de datos. Esto pareciera ser sólo un aspecto técnico, pero no lo es; también tiene una componente de negocio. Si le pedimos por ejemplo al departamento de finanzas que emita una factura, tenemos que enviarle los datos de facturación y estos en un formato estructurado como lo necesita el área de finanzas. Este formato estructurado se denomina en BPMN «data-mapping» o la interfaz de datos entre procesos superiores y subprocessos globales. Hablando sobre estos aspectos, usted ya se habrá dado cuenta que BPMN al hacer estas analogías trata de construir puentes entre el mundo de negocio y mundo técnico. ¿Cómo piensa BPMN lograr esta integración? BPMN intenta obligar al modelador a ser formal en su expresión. Si no fuese así, ¿cómo nos vamos a entender en nuestro mundo cada vez más integrado en sus procesos (complejos) y cambiantes?

2.8.3. Eventos sobrepuestos que impactan sobre el subprocesso

En secciones anteriores conocimos eventos intermedios que impactan durante la ejecución de una actividad (eventos sobreuestos). Estos mismos eventos pueden utilizarse de igual forma en subprocesos, hecho que abre una amplia gama de posibilidades en el modelamiento de procesos. Por ejemplo podemos considerar en el modelamiento ocurrencias inesperadas que impactan sobre un subprocesso en general. En nuestro ejemplo de la figura 2.83 dejamos de cocinar en el momento que nos entre una llamada invitándonos a cenar afuera.

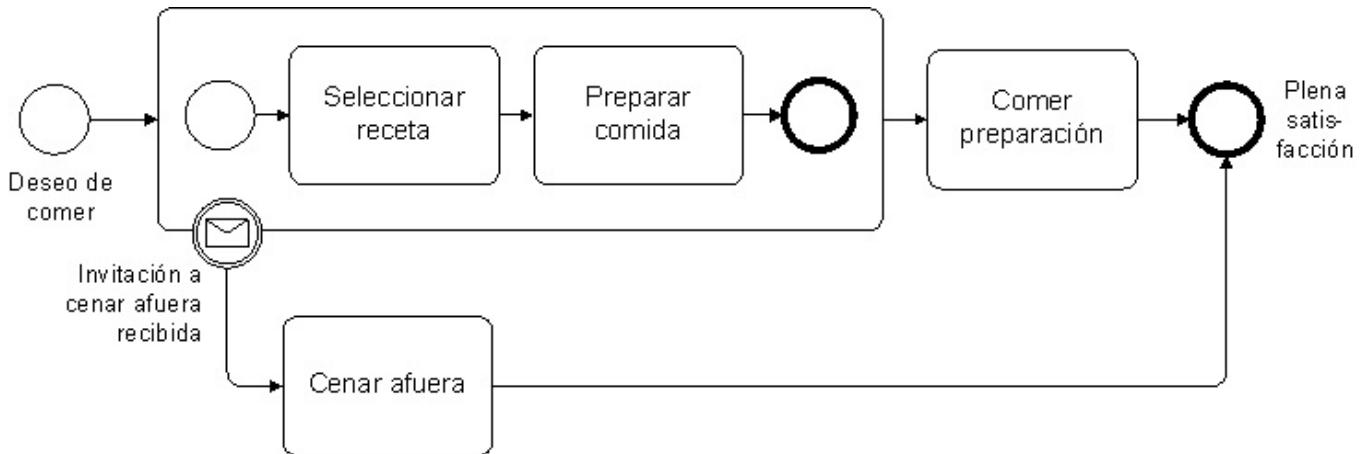


Figura 2.83: Caso con evento sobreuesto que interrumpe subprocesso completo

En relación a eventos del tipo de mensajería, de tiempo y de condición, el proceso superior siempre interrumpe el subprocesso si tiene que reaccionar a sucesos externos. En cambio en eventos del tipo de tratos especiales, como eventos de error, de cancelación o de escalación, el subprocesso informa al proceso superior sobre su acaecimiento para que se inicie un trato especial, como lo muestra el ejemplo de la figura 2.84 en que el artículo pedido ya no se fabrica, lo que interrumpe el proceso por falta de disponibilidad. La adquisición de artículos, es en este caso, un subprocesso global que reacciona ante la no disponibilidad de compra del artículo. Desde el punto de vista organizacional, es posible que en forma manual el ejecutivo de compra le comunique al ejecutivo de ventas que el artículo quedó fuera de serie y ya no está disponible. Interesante de notar en este contexto, es la posibilidad que se abre en BPMN a los procesos superiores de poder iniciar tratamientos especiales y diferenciados, por ejemplo en nuestro caso de informar al cliente y de borrar el artículo del catálogo. La decisión de si un subprocesso es abortado y como sigue el flujo la toma el proceso superior que lleva el control del proceso. BPMN nos permite de esta forma diseñar entornos de procesos modulares y flexibles ante el cambio.

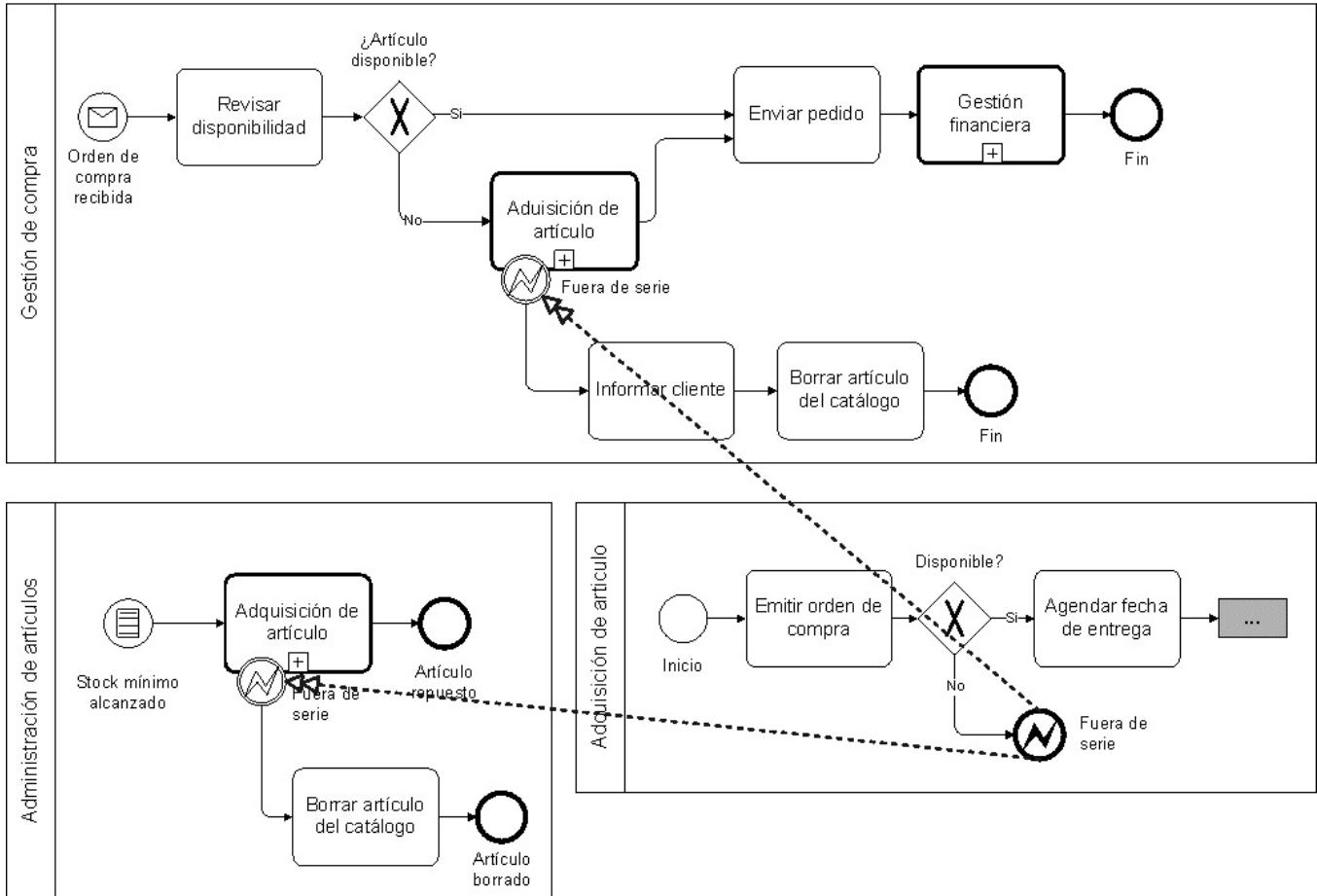


Figura 2.84: Caso con subprocesso que comunica un error al proceso superior

El evento de señal cumple una doble función en los subprocessos. La primera funcionalidad que puede cumplir es que el proceso superior capte una señal externa y se la transmita al subprocesso, igual que como un evento de mensaje puede reaccionar ante una señal externa durante la ejecución de un subprocesso. Pero el evento de señal también se utiliza preferentemente para que un subprocesso pueda comunicarle un mensaje al proceso superior, sin tenerlo que interpretar necesariamente como un error. La razón de esta funcionalidad especial del evento de señal en subprocessos se debe a que no podemos modelar este caso con eventos de mensajería, que sólo se utilizan en BPMN para el intercambio con participantes externos, es decir fuera de los límites del pool. La comunicación entre subprocessos (superiores e inferiores) no cuenta como comunicación externa o entre pools.

Como en otros casos, no estaríamos usando el evento de señal para lo que originalmente se concibió. En BPMN 2.0 contamos con una mejor alternativa, el evento del tipo escalación. En este caso el subprocesso puede comunicarse a través del evento de escalación con el proceso superior, sin que tenga que ser interpretado como un mensaje de error. Además, a partir de BPMN 2.0 el proceso superior puede captar el mensaje y procesarlo sin que sea necesario interrumpir el subprocesso, debido a que en la nueva versión tenemos a disposición los eventos intermedios del tipo «no interrupción» (ver figura 2.85).

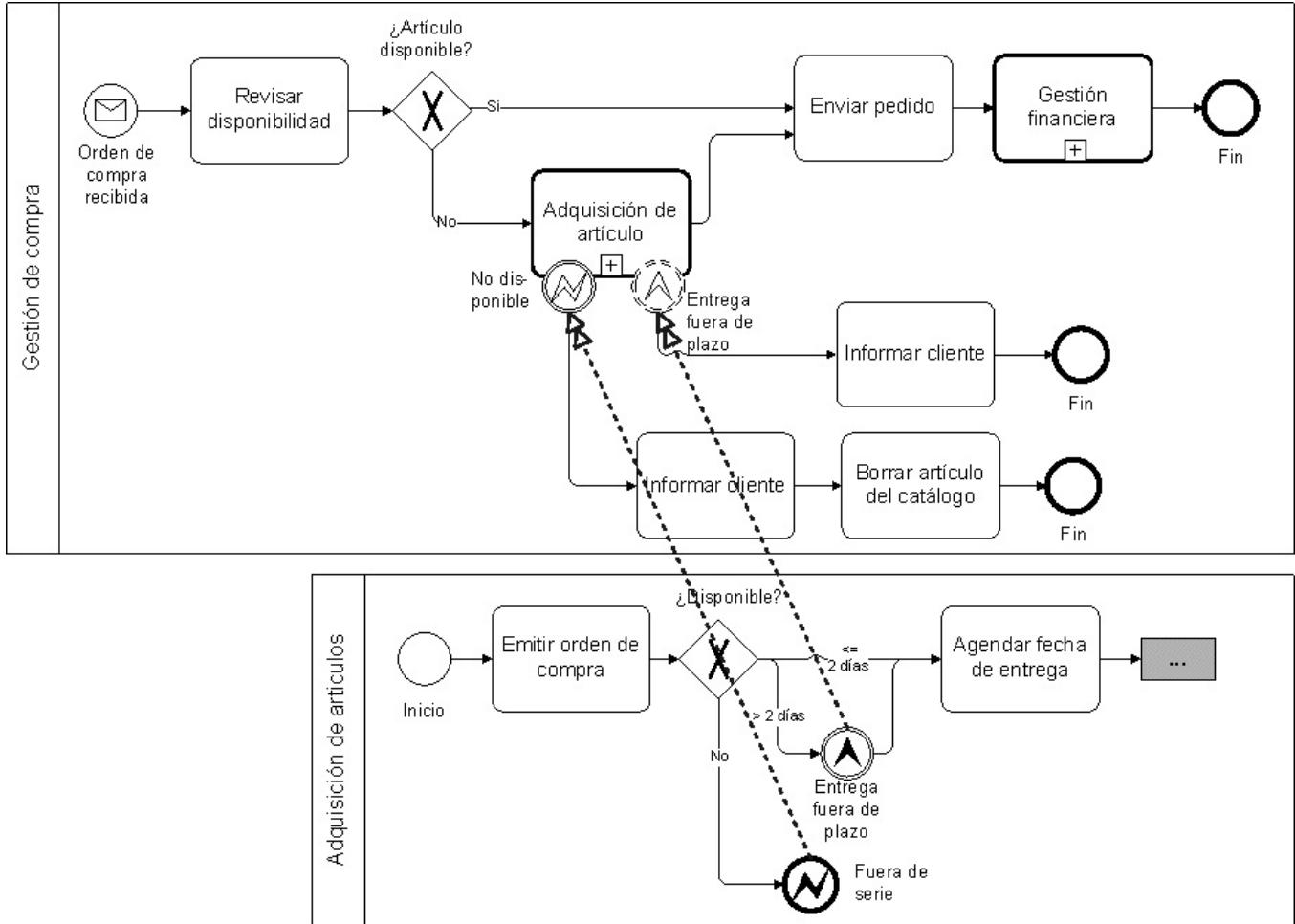


Figura 2.85: Caso con Evento de Escalación que comunica un Mensaje al Proceso Superior

2.8.4. Propiedades de Subprocesos

Las propiedades descritas en la sección 2.7.2 que pueden asumir actividades, como loop, instancias múltiples y compensación también aplican de igual forma para los subprocessos. De esta forma se pueden modelar loops complejos en forma más compacta como muestra la figura 2.86 en la cual ambos modelos son idénticos.

La propiedad especial denominada «ad-hoc» sólo puede utilizarse en subprocessos, símbolo que se reconoce como una «tilde» que se posiciona al centro inferior del subprocesso (ver figura 2.87). Las actividades o subprocessos contenidos en un subprocesso con propiedad de ad-hoc pueden:

no tener una secuencia determinada en su ejecución,
pueden ejecutarse n veces, o

se ejecutan algunas o ninguna de ellas.

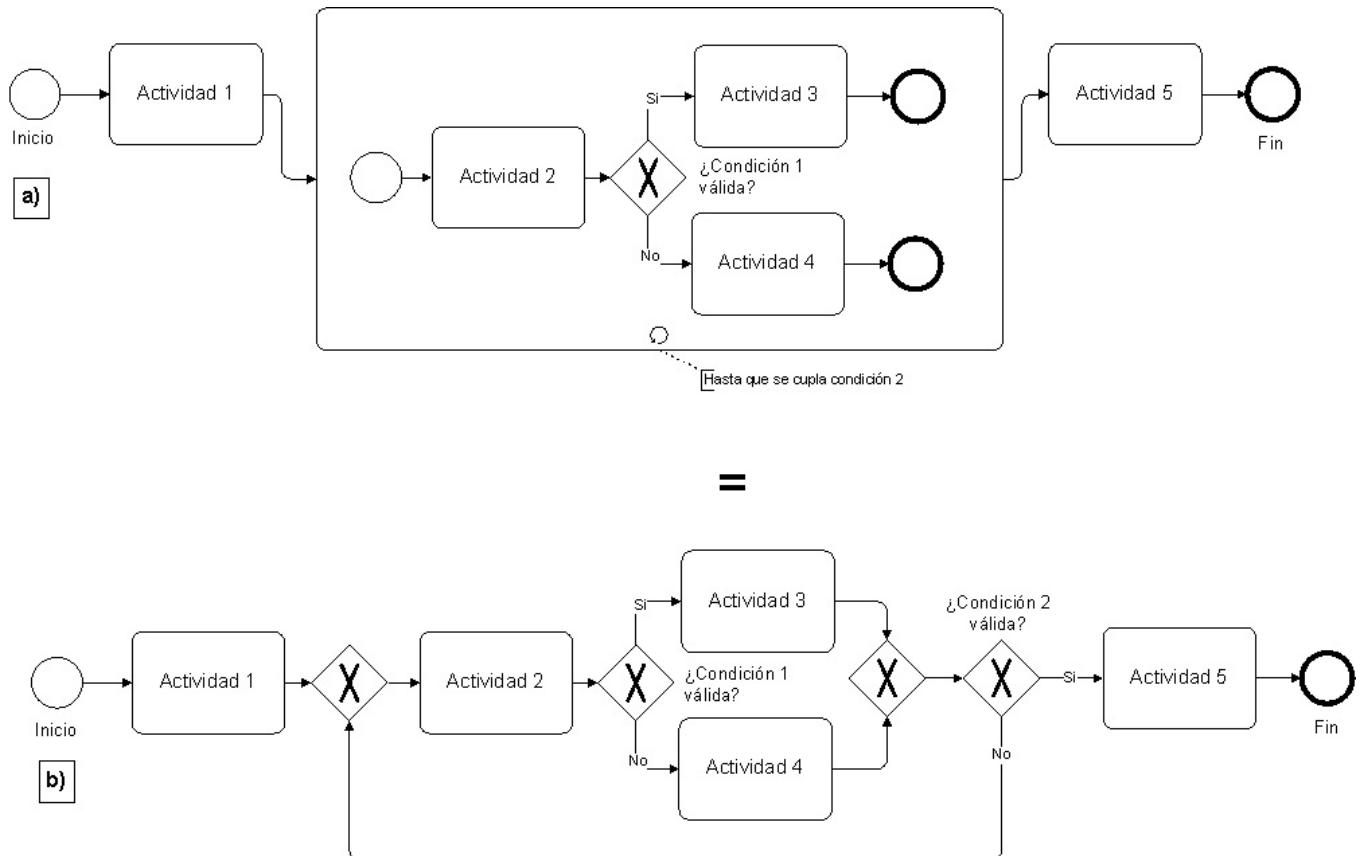


Figura 2.86: Equivalencia entre a) Subproceso con propiedad de loop y b) Flujo con actividades y XOR-Gateway que se comportan como loop

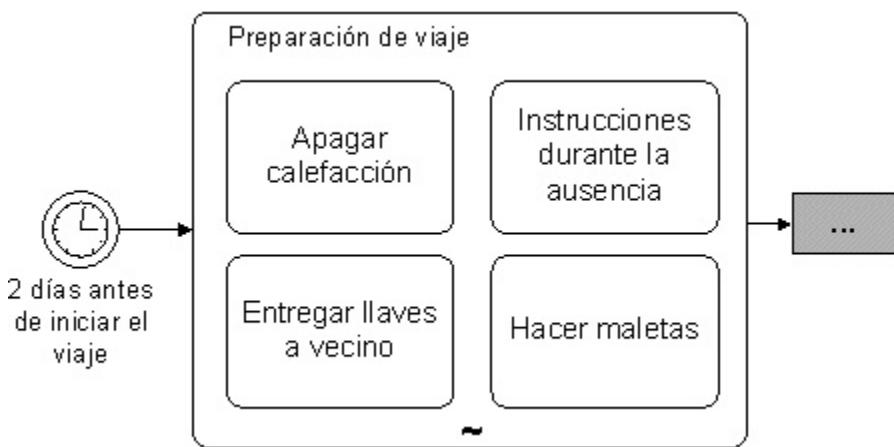


Figura 2.87: Caso con uso de subprocesso con propiedad Ad-Hoc

Figura 2.87: Caso con uso de subprocesso con propiedad Ad-Hoc

La decisión de cuales de estas opciones se realizan, las toma el responsable o encargado del subprocesso. De cierta forma esta componente nos lleva a pensar, que el modelamiento de procesos sería absurdo si tenemos este grado de libertad, pero la realidad nos muestra que existen muchos procesos, que requieren de este grado de libertad, porque dependiendo de la situación se necesita actuar de una forma u otra. Pensemos en los procesos clínicos de un hospital. Cuando usted entra a urgencia, el médico tiene que hacerle primero un diagnóstico y luego dependiendo de este resultado, lo va a derivar a una sección u otra. El flujo no está determinado y se va construyendo a medida que se va adquiriendo conocimiento sobre el caso. También

puede darse el caso que un proceso, sea un candidato a estandarizarse pero que aún no se encuentre bien estructurado, entonces ¿Cómo representamos la situación actual si no existe un flujo pre determinado?

BPMN 2.0 definió reglas sintácticas para el uso de subprocessos ad-hoc: Obligación: Deben existir al menos dos actividades o subprocessos

Opcional: Objetos de datos, flujos de secuencia, asociaciones, agrupaciones, flujo de mensajes, Gateways y eventos intermedios

Prohibido: Eventos de inicio y de fin, objetos de conversación y coreografía (definición de estos objetos más adelante)

Considerando no infringir estas reglas se pueden modelar con subprocessos ad-hoc, procesos con poca o ninguna estructuración (ver ejemplo en figura 2.88).

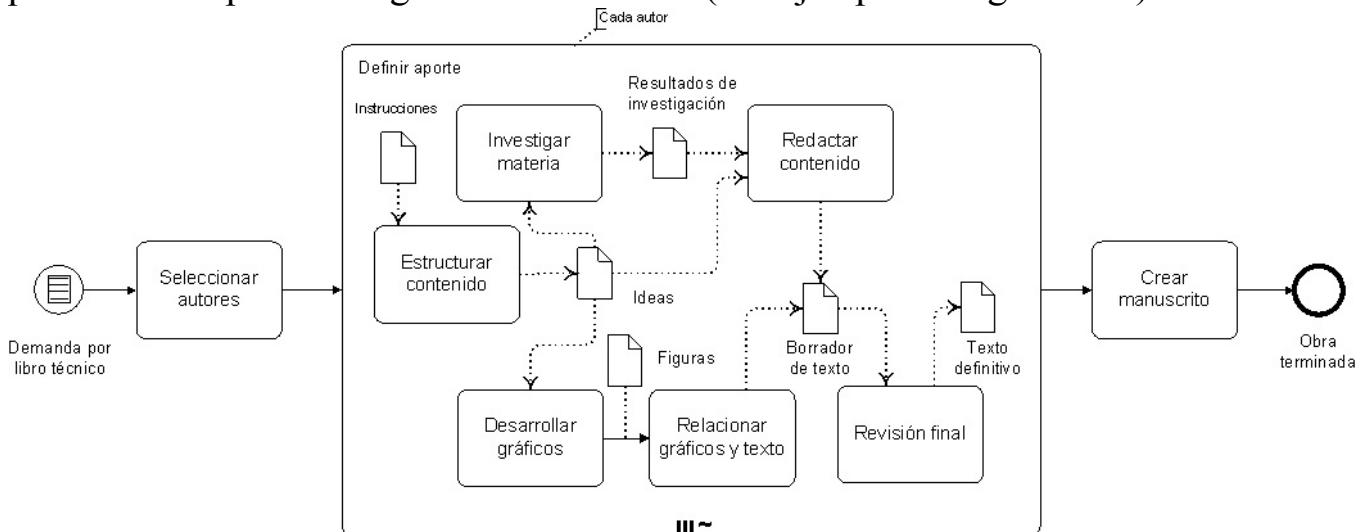


Figura 2.88: Caso con uso de subprocesso con propiedad Ad-Hoc con poca estructuración interna

2.8.5. Subproceso del tipo transaccional

Muchos procesos funcionan sólo bajo el principio «todo o nada», es decir o todos los pasos están correctos o no se debe hacer nada. En la sección 2.6.9 conocimos el evento de compensación, el cual impulsa una acción de reversar todo lo hecho anteriormente. En BPMN entendemos bajo el elemento de Transacción un subprocesso especializado que apoya las funcionalidades de procesos transaccionales. La figura 2.89 muestra en forma ejemplar como funciona un proceso transaccional: Vamos a suponer que queremos visitar en nuestras vacaciones a nuestros familiares que viven en otro país. Una vez tomada esta decisión vamos a planificar y preparar nuestro viaje. Primero vamos a coordinar con ellos una fecha y la duración de la estadía. Luego vamos a reservar pasajes de vuelo en una línea aérea económica y reservar una habitación en un hotel cercano a nuestros familiares, a pesar que ellos insisten que nos alojemos en su casa. En realidad pensamos en nuestra independencia, además de no comprometernos o abusar de ellos. Por último tenemos que solicitar la fecha de vacaciones a nuestro jefe. Si todo resulta bien

podemos emprender nuestro viaje, pero ¿qué hacemos si el hotel está copado y no encontramos otro alternativo debido a la sobre demanda de la época, o si nuestro jefe no nos aprueba la fecha de vacaciones solicitada? En este caso tenemos que abortar nuestra preparación del viaje. En BPMN existe un símbolo que sólo se puede utilizar para cancelar transacciones (cruz blanca al borde del subprocesso). Si una transacción es cancelada, se disparan automáticamente todos los eventos de compensación que son necesarios. Entonces le avisamos a nuestros familiares que no podemos viajar, el ticket de vuelo lo perdemos porque las condiciones tarifarias no permiten devolución de dinero (compensación no posible) y cancelamos la reserva de hotel. Luego de compensar todas las actividades posibles, el evento de transacción cancelada nos lleva nuevamente a la planificación de viaje y buscamos otra fecha alternativa.

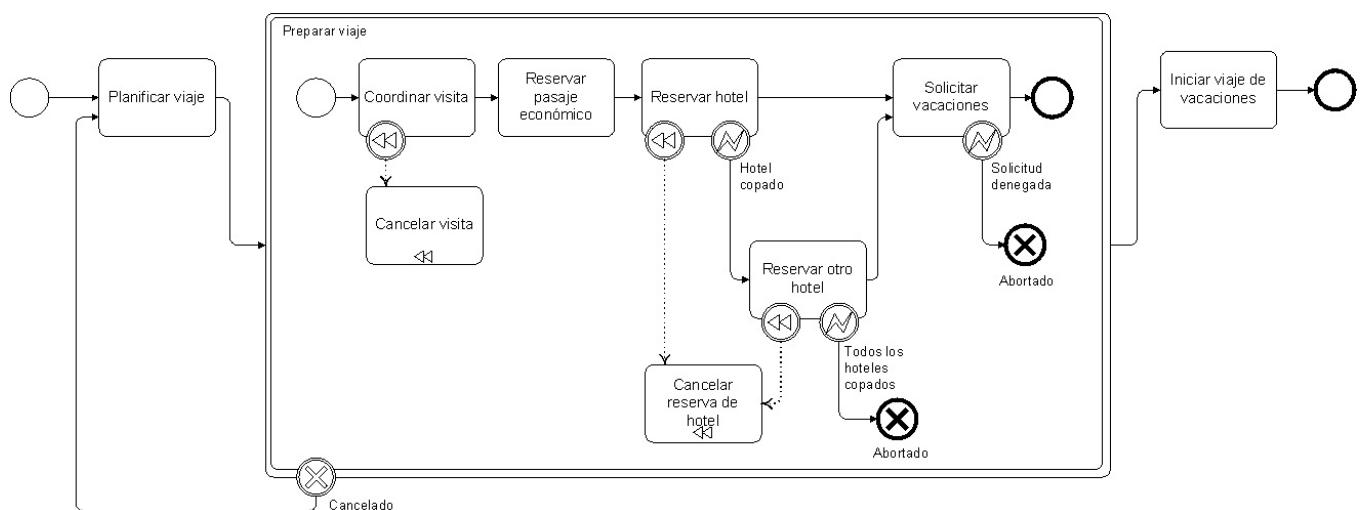


Figura 2.89: Caso con uso de subprocesso expandido del tipo Transacción

2.8.6. Subproceso impulsado por eventos

En BPMN 2.0 se introdujo un nuevo objeto llamado subprocesso impulsado por eventos (en inglés: event-subprocess). Este tipo de subprocesso debe encontrarse dentro de un proceso o subprocesso y se reconoce por su delimitación interpunteada. Este proceso es gatillado por un sólo evento de inicio y sólo puede ser iniciado durante la ejecución del proceso que lo contiene. El subprocesso impulsado por evento no puede interrumpir por el evento el proceso superior. Este comportamiento lo reconocemos en el tipo de evento de inicio que lo contiene. En el caso de subprocessos del tipo no interrupción, pueden ser instanciados en forma repetitiva o tantas veces como ocurran, hasta que termine la ejecución del evento superior.

En el ejemplo de la figura 2.90 mostramos como se comportarían estos dos tipos de subprocessos. Invitamos a algunos amigos a cenar e iniciamos el subprocesso de preparación de la cena. Seleccionamos una receta y empezamos a preparar la comida. Mientras estamos cocinando suena el teléfono y se anuncia la visita de otro huésped. Como no nos complicamos con la llegada de un nuevo comensal

preparamos una porción más y ponemos un puesto más en la mesa. Pero como no somos muy experimentados en cocinar, durante la preparación se nos quema la comida, hecho que conduce a un error fatal en todo nuestro proceso e interrumpe la preparación. Pero como somos precavidos, habíamos previsto introducir para estos casos un subprocesso impulsado por evento, que nos da una salida y pedimos comida preparada a un servicio externo. Cualquiera sean los resultados, finalmente nos sentamos junto a nuestros invitados a cenar.

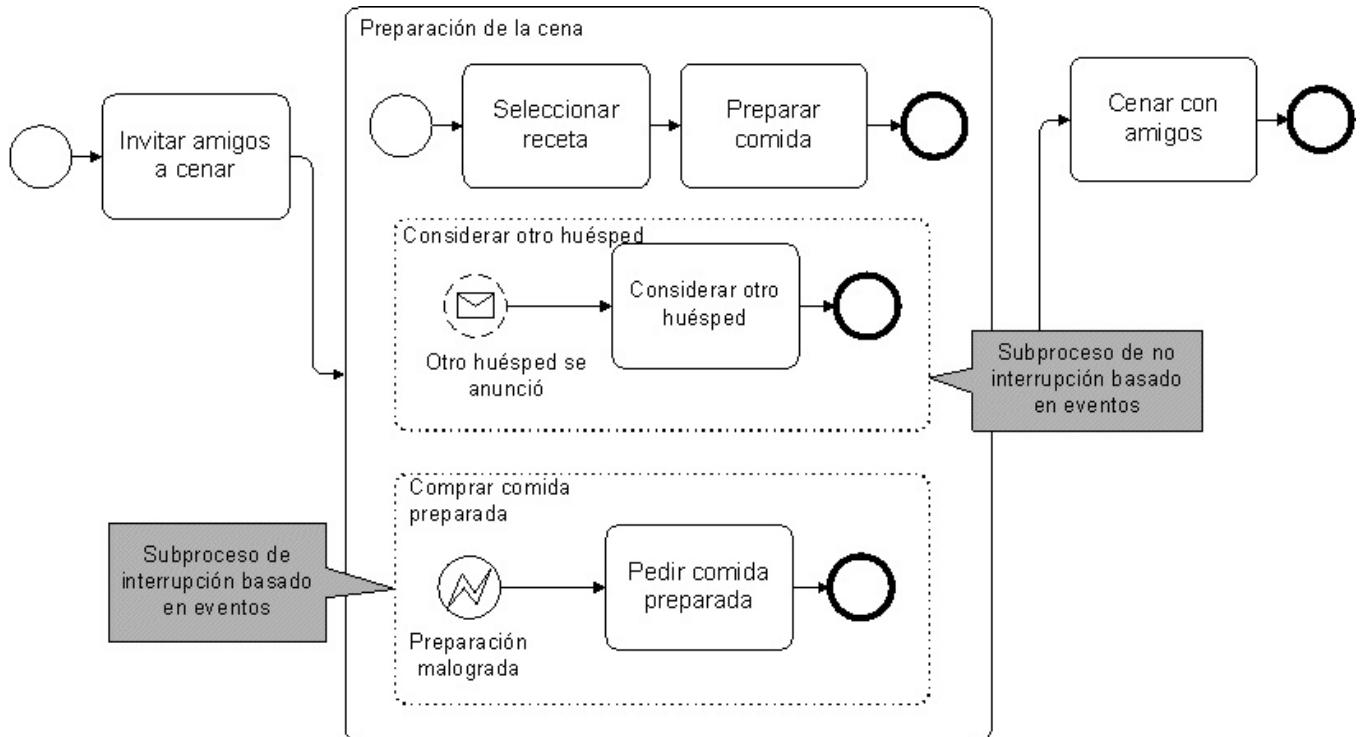


Figura 2.90: Ejemplos de subprocessos impulsados por eventos

La figura 2.91 muestra este mismo proceso, pero con los subprocessos cerrados. La representación es la misma, sólo que el subprocesso cerrado lleva el signo (+) en el objeto y a la izquierda arriba se reconocen los tipos de eventos que marcan los subprocessos impulsados por eventos (de interrupción o no interrupción).

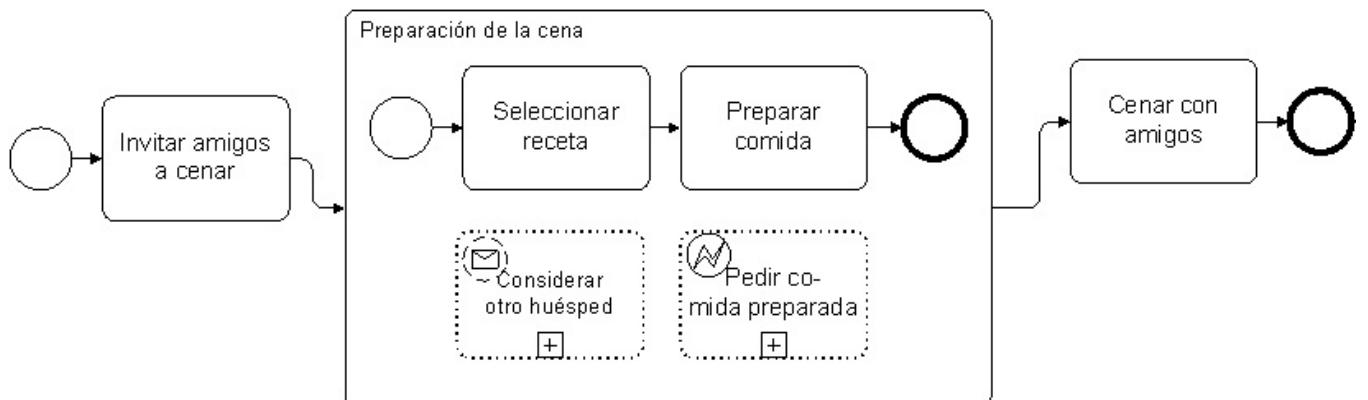


Figura 2.91: Caso con uso de subprocessos impulsados por eventos

Tipos de eventos que pueden gatillar subprocessos de eventos del tipo no interrupción

pueden ser:
de mensajería
de tiempo
de escalación
de condición
de señal
múltiples
múltiples paralelos

A los anteriores pueden utilizarse para subprocessos de eventos del tipo de interrupción:

de error
de compensación

En BPMN existen muchas formas de representar un mismo caso. ¿Cree usted que podemos modelar este caso, sin subprocessos impulsados por eventos? Si se puede y nosotros lo demostramos en el modelo que encuentra en la figura 2.92. Desde el punto de vista de la lógica, representan ambos modelos (figuras 2.90 y 2.92) casi la misma funcionalidad, con la salvedad que los casos de excepción no se procesan dentro del proceso superior sino en forma separada. Entonces para subprocessos globales se dan las siguientes consecuencias:

Si el subprocesso está inserto en otra área de responsabilidad, se encargan otros participantes de la ejecución de los subprocessos o del tratamiento de los eventos. Si la responsabilidad del subprocesso recae en el mismo proceso superior, tiene que el mismo pool o participante hacerse cargo de la ejecución del subprocesso.

Como el subprocesso es global y por lo tanto reutilizable, recae la decisión de como reacciona el subprocesso en el proceso superior que lo invoca.

Los subprocessos globales no pueden acceder directamente a los datos del proceso superior; se requiere de un mapping, lo que no sería necesario en un subprocesso de evento no global.

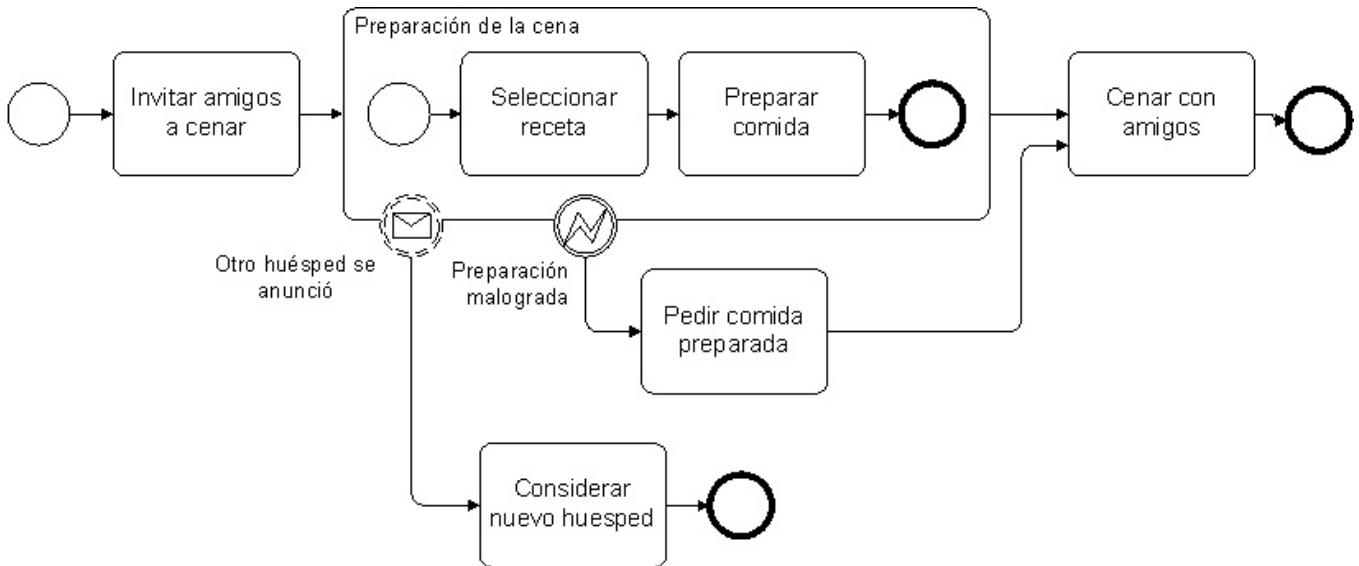


Figura 2.92: Caso con utilización de Eventos sobrepujantes en el subproceso como modelamiento alternativo a figura 2.90

2.9. Pools y Fluxos de Mensajes 2.9.1. El dirigente y su orquesta

En la sección 2.5 explicamos como se asignan responsables a los procesos en las actividades o subprocessos por medio de lanes. Estos lanes siempre se encuentran dentro de un pool (contenedor), el cual representa al mismo tiempo los límites de un proceso de principio a fin, pero además cumple otra función importante en BPMN: El pool representa una instancia invisible que dirige y controla el flujo de actividades en el proceso. En la figura 2.93 podemos apreciar como el pool toma el rol de «dirigente», por ejemplo cuando Roberto termina su actividad 1, el dirigente le asigna la actividad 2 a Falko y así sucesivamente. El concepto se utiliza en analogía a una orquesta de música, en la cual el dirigente guía y controla a cada uno de los músicos, razón por la cual en BPMN también se habla de «orquestación».

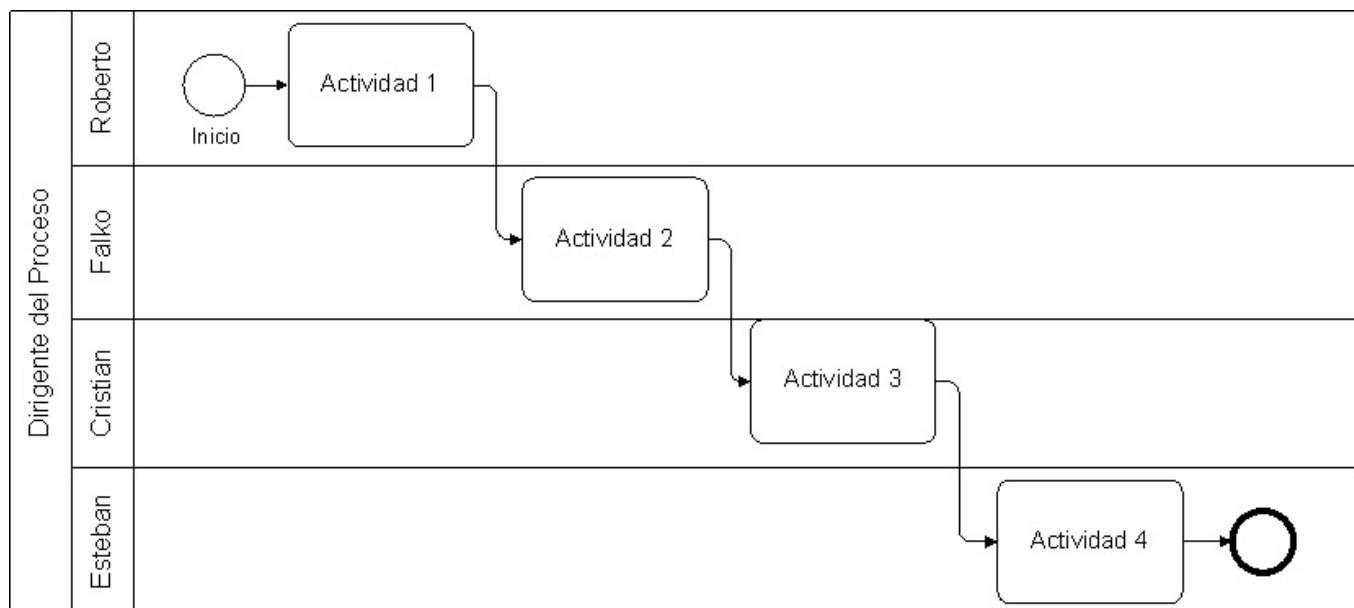


Figura 2.93: Actividades y responsables de actividades

Aunque parece sencillo y entendible, muchos analistas desarrollan sus modelos pensando que tienen que modelar la funcionalidad de orquestación, como lo muestra la figura 2.94. Ellos piensan que no existe un dirigente y que los participantes tienen que coordinarse, pero la coordinación dentro de un pool en BPMN no es necesaria y se muestra a través del flujo de secuencia. Si se requiere de coordinación para BPMN sólo puede ocurrir fuera de un pool o mejor dicho entre pools, que al mismo tiempo representan participantes.

Si se hace necesario una coordinación entre participantes, BPMN obliga a separar los pools y la comunicación entre ellos se lleva a cabo a través de flujos de mensajes. Entonces tendríamos en el ejemplo que ilustra la figura 2.95 cuatro dirigentes, cada uno con su propio mini-proceso y su propio flujo de control. Entre ellos no pueden hacer otra cosa que intercambiar información a través de flujos de mensajes. Es posible que en un proceso dependa de un mensaje externo para que pueda continuar, pero eso lo define el propio proceso (pool) dentro de su lógica.

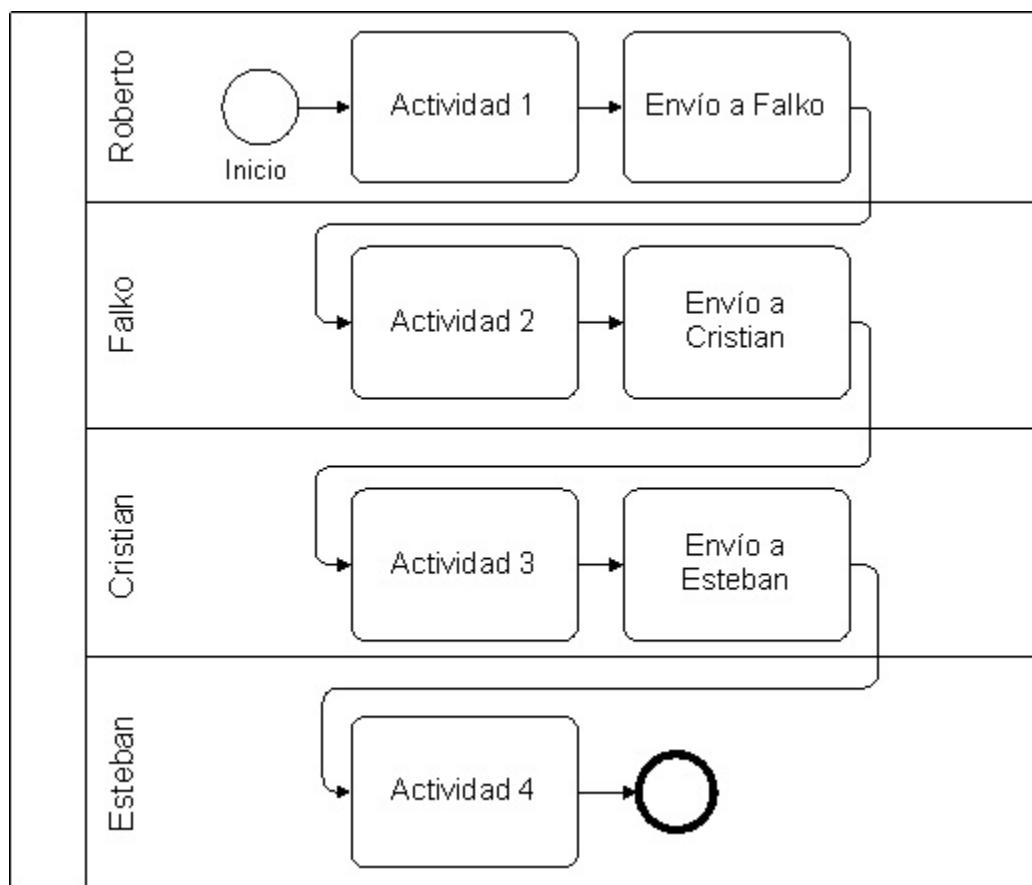
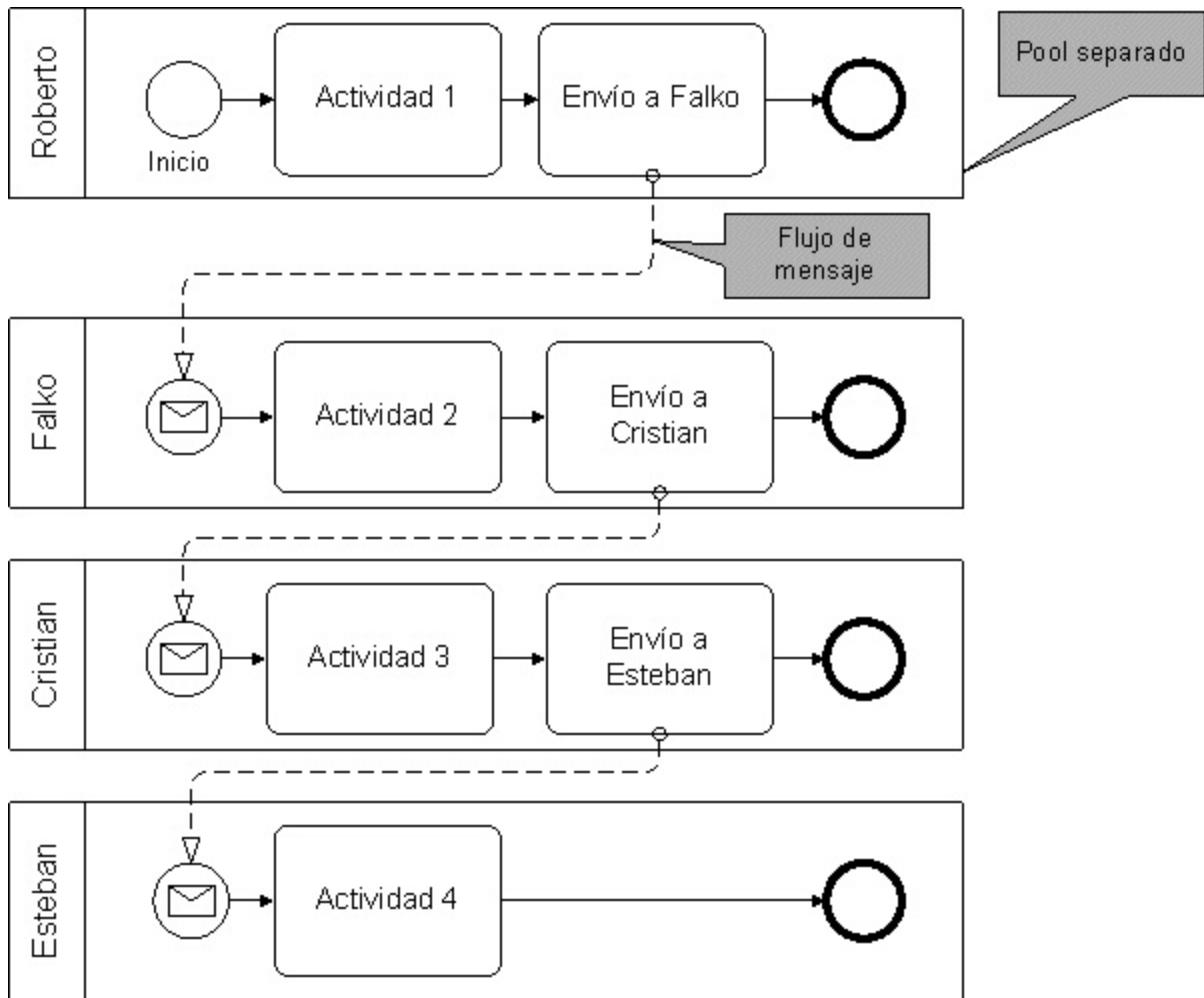


Figura 2.94: Flujo sin nombre de Pool



Figura

2.95: Flujo con cuatro participantes en Pools propios

Es posible que usted no se sienta muy cómodo con este principio de modelamiento, porque en otras técnicas de modelamiento no se interpreta así. En muchas ocasiones no es necesario separar a todos los participantes para representar una determinada lógica en un proceso, eso va a depender fundamentalmente si el «dirigente» tiene el control sobre ellos. Si un pool o dirigente no tiene control sobre un participante, entonces si tiene que obligadamente separarlo y representarlo como un pool propio, por ejemplo clientes y proveedores.

¿Por qué BPMN introdujo este principio de separar los participantes a pools propios para representar la lógica en sus diagramas? Si usted se acuerda, el objetivo principal que tienen los autores del BPMN en mente es la automatización de los procesos a partir de los diagramas. Usted a lo mejor ha escuchado hablar de orquestación de servicios en el contexto de arquitecturas orientadas a servicios (SOA). BPMN persigue el mismo principio, con la única diferencia que la orquestación de servicios es completamente automática y en el caso de un motor de workflow interviene principalmente el ser humano (human-workflow).

La siguiente pregunta que se da en este contexto es ¿qué pasa con aquellos procesos que por diversas razones no van a ser automatizados? o lo que también sucede mucho

en la práctica, ¿qué parte de los procesos de negocio son procedimientos manuales? Justamente la separación de pools permite separar la lógica de los procesos manuales de los procesos automatizados, lo que es una gran ventaja ya que deja en claro qué parte del proceso pasará al diseño técnico de implementación y qué parte pasará a ser procedimiento manual. Usted tiene la libertad de modelarlo como quiera; puede modelar el proceso dentro de un gran pool con lanes (como en la figura 2.94), pero a más tardar en el diseño técnico tendrá que separar estos aspectos, lo que sin duda alguna significa doble trabajo, mayor coordinación y separación entre las capas. Por otro lado si queremos lograr una mejor integración o alineamiento entre la capa de negocio y la de TI, el modelo de (separación de pools) que propone BPMN aporta justamente a lograr este objetivo. En los próximos capítulos vamos a mostrar los beneficios que tenemos al aplicar este concepto muy típico de la estructura del BPMN.

2.9.2. Reglas de aplicación

Si usted trabaja con el concepto de pools y flujos de mensajes, puede modelar las siguientes situaciones (ver figura 2.96):

1. Flujos de mensajes que vienen de afuera e impulsan eventos de mensajes dentro del pool
2. Actividades que envían y/o reciben flujos de mensajes
3. Eventos de mensajes disparados por el proceso que provocan un flujo de mensaje hacia afuera
4. Subprocesos abiertos que reciben o envían flujos de mensajes

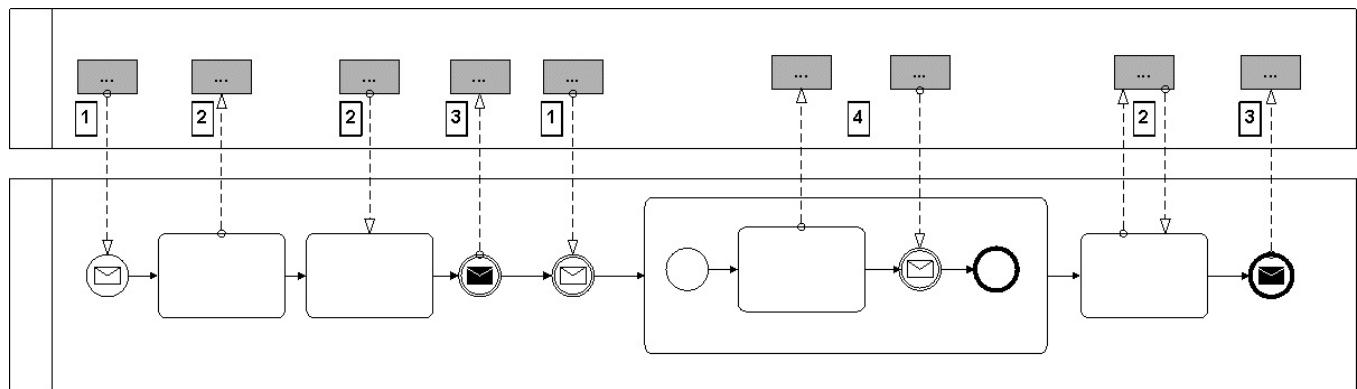


Figura 2.96: Elementos permitidos para el intercambio de información (flujos de mensajes) entre pools

El empleo de las siguientes relaciones infringe las reglas de BPMN y no nos está permitido usarlas (ver figura 2.97):

1. Flujos se secuencia que traspasan los límites de un pool
2. Flujos de mensajes con flechas en ambos extremos de la misma

3. Eventos conectados a flujos de mensajes que no son del tipo de mensajería
4. Flujos de mensajes dentro de un pool
5. Gateways con flujos de mensajes
6. Eventos que envían y reciben flujos de mensajes

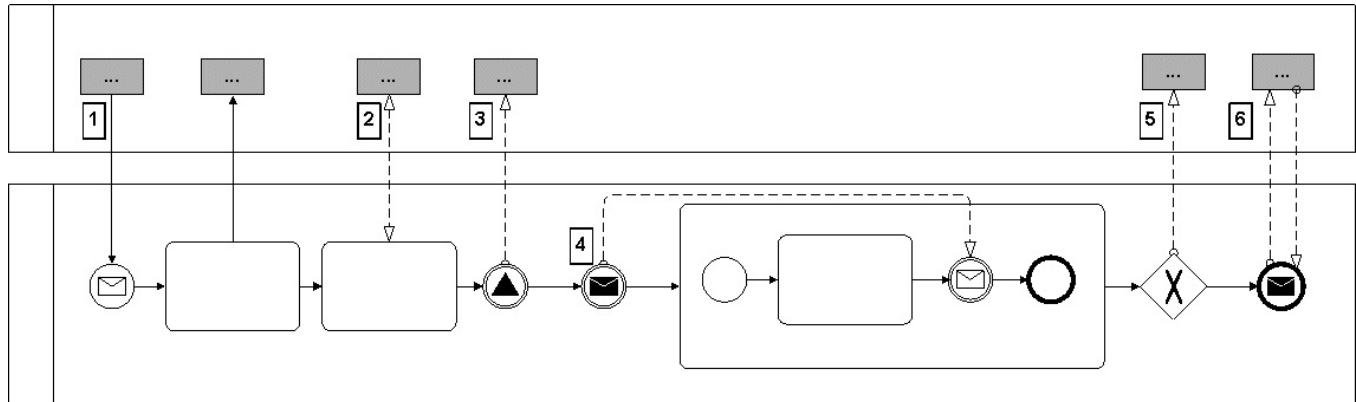


Figura 2.97: Elementos no permitidos relacionados con flujos de mensajes

2.9.3. El arte de la colaboración

Con el ejemplo de la figura 2.98 describimos el modelamiento del caso de EventGateways, es decir nuestro proceso de pedido de pizza. Ahora queremos agregar la vista del proceso del proveedor de pizza, es decir de la pizzería. Usted encuentra el modelo de este proceso en forma resumida en la figura 2.99. Cuando la pizzería recibe una orden, el hornero prepara la pizza pedida, el servicio de entrega la lleva al cliente y cobra el pedido al entregarla.

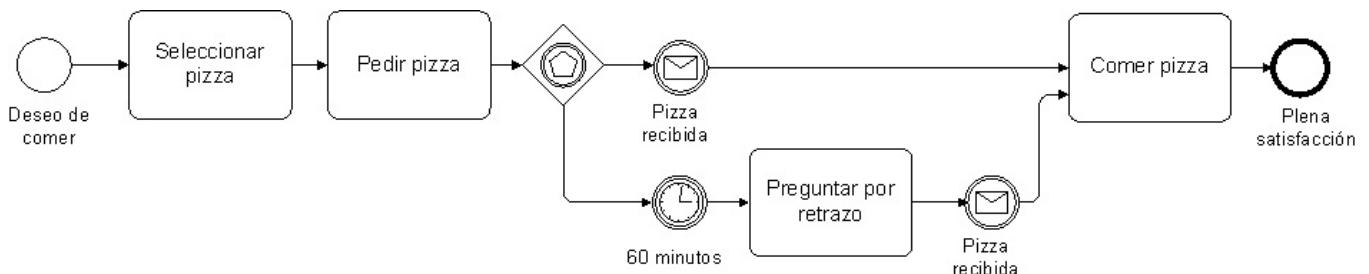


Figura 2.98: El proceso de pedido de pizza desde el punto de vista del cliente

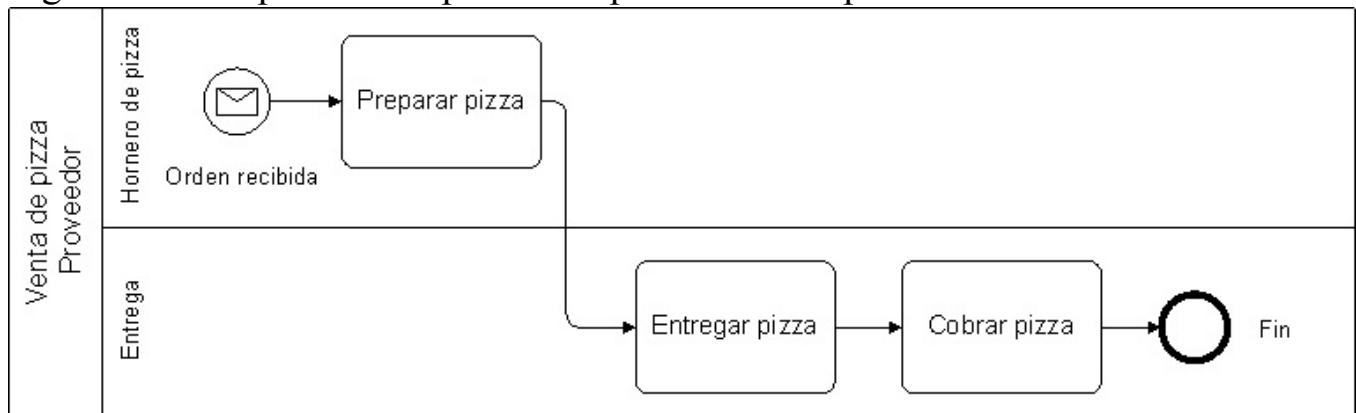


Figura 2.99: La entrega de pizza desde el punto de vista del proveedor

En el siguiente paso, lo que queremos hacer es relacionar estos dos procesos del

cliente y el proveedor. Primero podemos modelarlo en un pool con diferentes lanes, entonces obtendríamos un proceso como muestra la figura 2.100, pero esta representación no es del todo correcta, porque existen eventos que se refieren a la interacción entre ambos participantes, por ejemplo el esperar el pedido o la cobranza. También existen actividades que se desenvuelven en forma independiente de los participantes, como el preparar la pizza (proveedor) y la cena en casa del cliente.

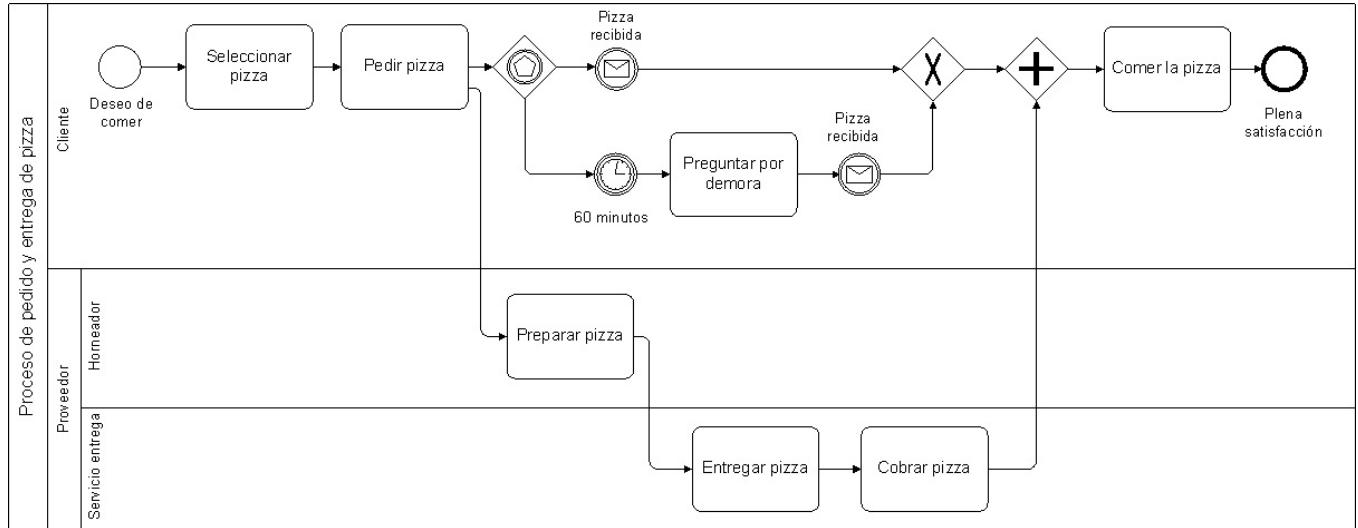


Figura 2.100: El proceso de negocio de pedido y entrega de pizza representado en un pool

Si modelamos el proceso con dos pools por separado obtendríamos un modelo como lo muestra la figura 2.101. Cada participante representa su propio proceso y se modelan los puntos de interacción entre ambos. A este tipo de representación se le llama en BPMN diagramas de colaboración, es decir muestra la colaboración entre dos procesos independientes.

No confunda procesos independientes con procesos de negocio. Un proceso de negocio en BPMN es representado por uno o más diagramas de colaboración con varios procesos independientes.

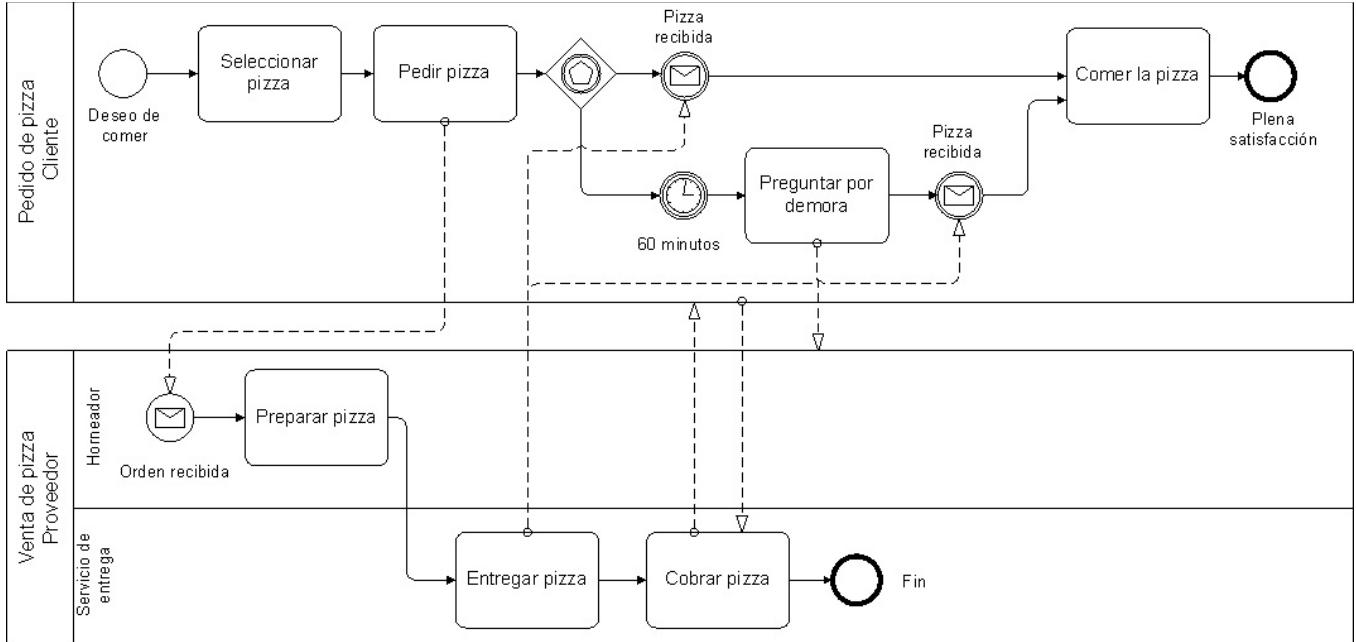


Figura 2.101: El proceso de negocio del pedido y entrega de pizza modelado en dos pools

En dos casos de nuestro ejemplo, existen flujos de mensajes que no terminan en una actividad sino en el borde del pool: el primer caso cuando preguntamos por el atraso y el segundo relacionado con la cobranza. En el primer caso se debe ésta representación a que el reclamo no «influye» en el proceso del proveedor. Es posible que el proveedor nos de una respuesta o que a lo mejor apure la preparación, pero la secuencia del proceso en general no se altera (hornear, entregar, cobrar). El segundo caso, de acuerdo a nuestras suposiciones, si tiene un déficit de modelamiento, porque de seguro que no se entregará la pizza si no la pagamos. Esta actividad «de pagar» falta en nuestro modelo y tenemos que incluirla. La figura 2.102 muestra el modelo correcto y ahora podemos relacionar el flujo de cobranza con nuestra actividad de pagar, en vez de conectarla al borde del pool de cliente.

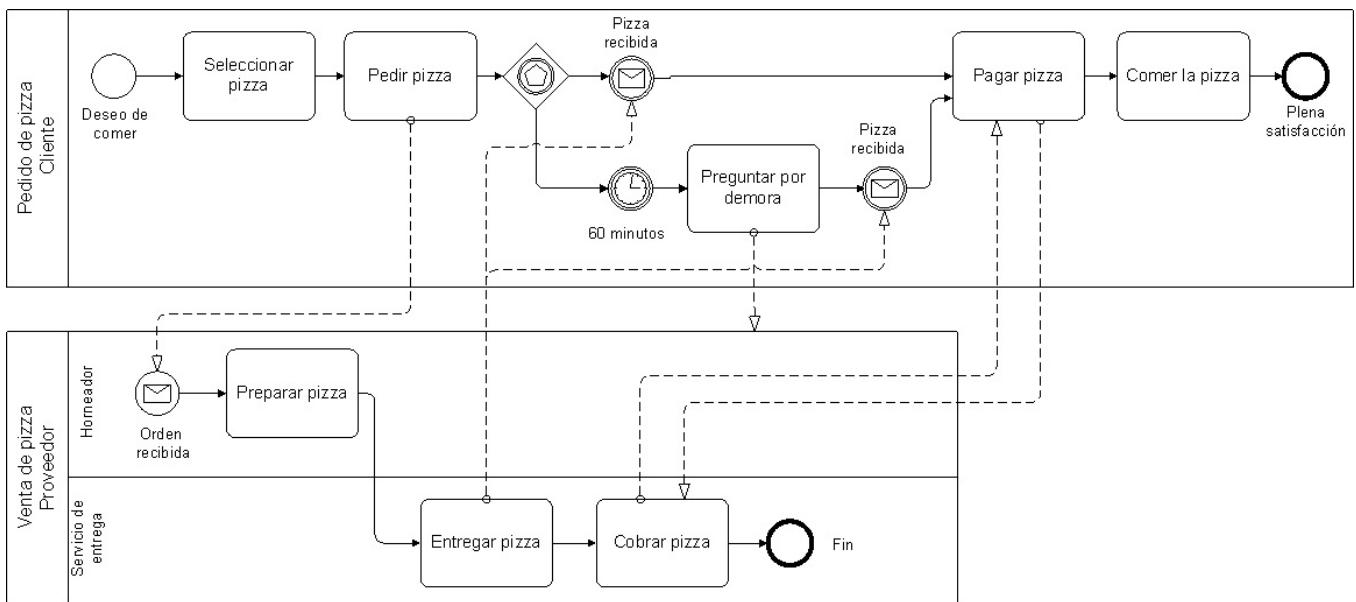


Figura 2.102: El proceso de negocio de pedido y entrega de pizza modelado

correctamente

2.9.4. Pools cerrados

En muchas ocasiones cuando modelamos procesos en la realidad no sabemos como funcionan los procesos internos de nuestros participantes. Por lo general tenemos algunas nociones de como funcionan los procesos externos, pero no conocemos su lógica, sólo tenemos dominio y conocimiento en detalle sobre nuestro propio proceso, pero para coordinar todos los puntos de interacción con los participantes no es necesario conocer la lógica de sus procesos.

La interfaz se modela por medio de los «flujos mensajes» en BPMN. Si adoptamos en nuestro ejemplo el rol del cliente, esperamos de nuestro proveedor que:

nos reciba la orden del pedido de la pizza,
haga entrega del pedido
cobre su servicio y
mantenga un servicio de atención a cliente.

En realidad no nos debiera de interesar como está organizado el proceso interno del proveedor. Seguramente tampoco nos vamos a enterar cuando se le acaben los ingredientes, que irá a buscar la pizza a otra sucursal, etc., pero eso es su problema, a nosotros sólo nos interesa que funcione el servicio. Cuando nos encontramos con situaciones como estas, lo mejor es cerrar el pool del participante y relacionar el flujo de mensajes con el borde del pool (ver figura 2.103).

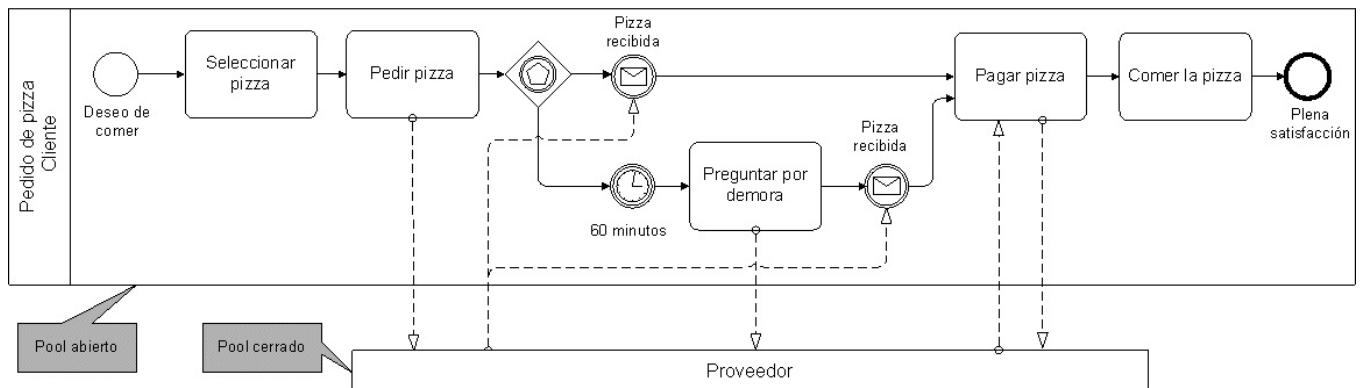


Figura 2.103: Caso con uso de Pool abierto y Pool cerrado

De acuerdo a la especificación podemos incluso cerrar todos los pools y sólo ver el intercambio de información entre los participantes, si es que le damos nombres a los flujos, como lo muestra la figura 2.104. La interacción entre pool cerrados nos puede servir para recibir una vista general del proceso de negocio, pero también tiene la desventaja que no podemos reconocer condiciones y dependencias de flujos alternativos, como en nuestro ejemplo el reclamo por el atraso de la entrega de la pizza. BPMN 2.0 ha dado una solución a este problema y se ha desarrollado justamente un nuevo tipo de diagrama llamado «diagrama de coreografía» que

muestra las dependencias de los flujos entre los participantes (ver sección 2.13).

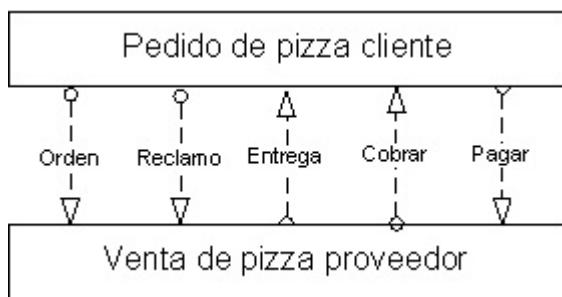


Figura 2.104: Intercambio de información entre dos pools cerrados

2.9.5. Pool con instancias múltiples

En las secciones 2.7.2 y 2.8.4 tratamos la posibilidad de marcar las actividades y los subprocesos con la propiedad de «múltiple». Esto significa que estos elementos pueden instanciarse en forma múltiple. Este principio también está disponible a partir de BPMN 2.0 para pools. Como un pool representa siempre sólo un participante se le llama a este elemento «pool con instancias múltiples». En la figura 2.105 vemos en un ejemplo como se puede emplear. En este proceso interactúan tres participantes, el cliente, la agencia y el proveedor. Cuando nace una instancia en el pool de la agencia es porque un cliente envió una orden de compra. La agencia envía cotizaciones a varios proveedores, es decir la actividad múltiple está relacionada con el pool múltiple; cada solicitud de cotización es enviada a distintos proveedores. En el momento que se hayan recibido todas las ofertas, se evalúan y se escoje una de ellas.

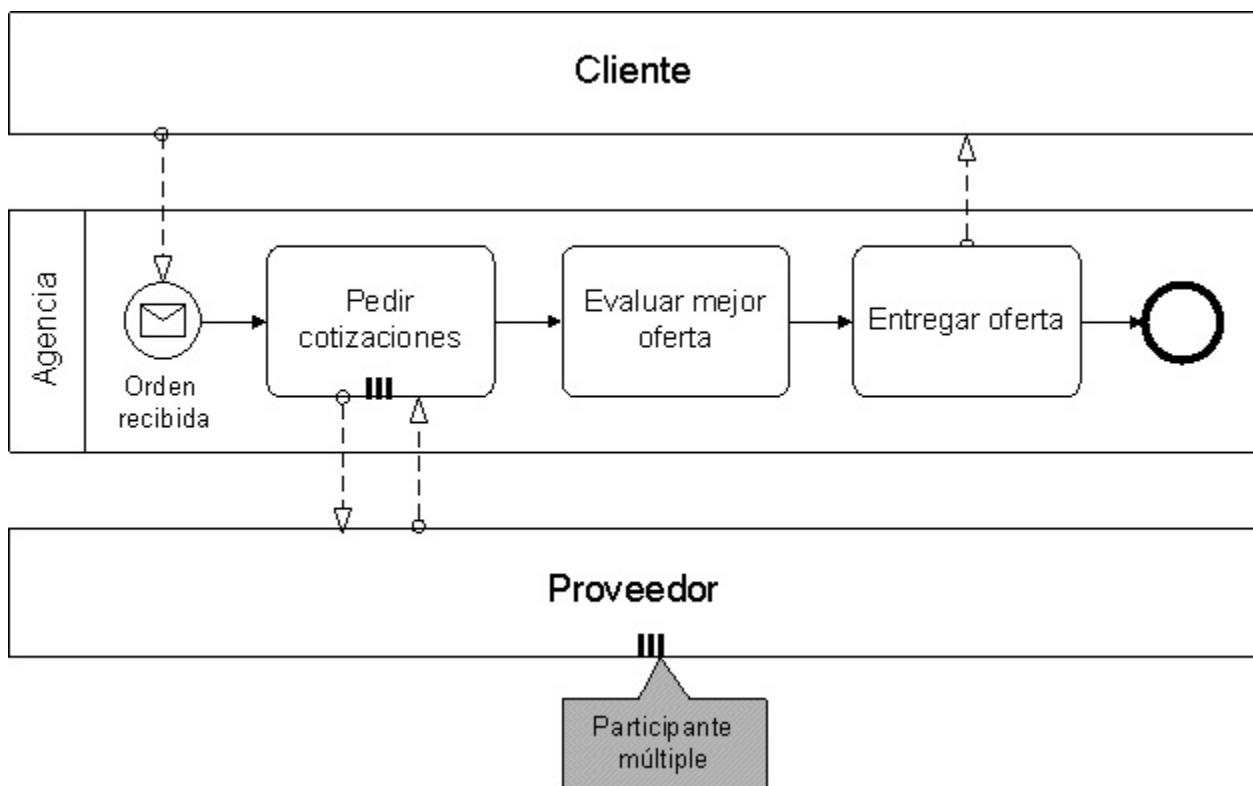


Figura 2.105: Caso con uso de Pool cerrado con propiedad de instancias múltiples
Como puede apreciar el pool múltiple puede ser muy útil en diagramas de colaboración en que interactúan varios procesos con sus respectivos participantes que pueden ser uno o muchos según el caso de negocio.

2.10. Objetos de datos en BPMN 2.0

BPMN se centra en la representación de la lógica del negocio con ayuda del flujo de secuencia, a través de las dependencias temporales y lógicas, combinando el flujo de secuencia con actividades, Gateways y eventos. Todos los demás aspectos que puedan tener relevancia para la descripción de los procesos son de menor importancia para la notación, como es el caso de representar datos o documentos que se utilizan en el proceso, pero usted puede considerar estos aspectos en los diagramas modelando con «objetos de datos». Un objeto de dato en BPMN representa cualquier tipo de información, como archivos físicos en papel, documentos, tablas y en general información electrónica.

Los objetos de datos se relacionan a través del objeto asociación con elementos de flujo o directamente al flujo de secuencia. Objetos de datos también pueden ser portador de un estado, el cual se muestra en BPMN con corchetes. Los estados que puede adquirir un objeto de datos son:

creado
por revisar
revisado
corregir
corregido
rechazado
aceptado

En el ejemplo abstracto de la figura 2.106 vemos la siguiente interacción entre los pools 1 y 2: Pool 2 crea en la actividad 2.1 el objeto de datos 2.1 con el nombre «Estado A», que se representa visualmente con el elemento de asociación y flecha dirigida hacia el objeto. Este mismo objeto de datos se necesita como input en la actividad 2.3, razón por la cual asociamos con flecha dirigida hacia la actividad este objeto. Además recibe la actividad 2.3 el output de la actividad 2.2. Como estas dos actividades están directamente relacionadas a través del flujo de secuencia, podemos desistir de la asociación dirigida en el objeto de datos 2.2 y relacionarlo directamente con el flujo de secuencia, es decir una forma más compacta de representar una relación input/output de un objeto de datos entre dos actividades. Luego la actividad 2.3 transforma el «Estado A» del objeto de datos 2.1 en «Estado B» y envía la información por medio de un flujo de mensaje al Pool 1. En BPMN 2.0 este envío ya no se representa por un objeto de datos, sino como mensaje

representado por un sobre de carta que se asocia al flujo de mensaje.

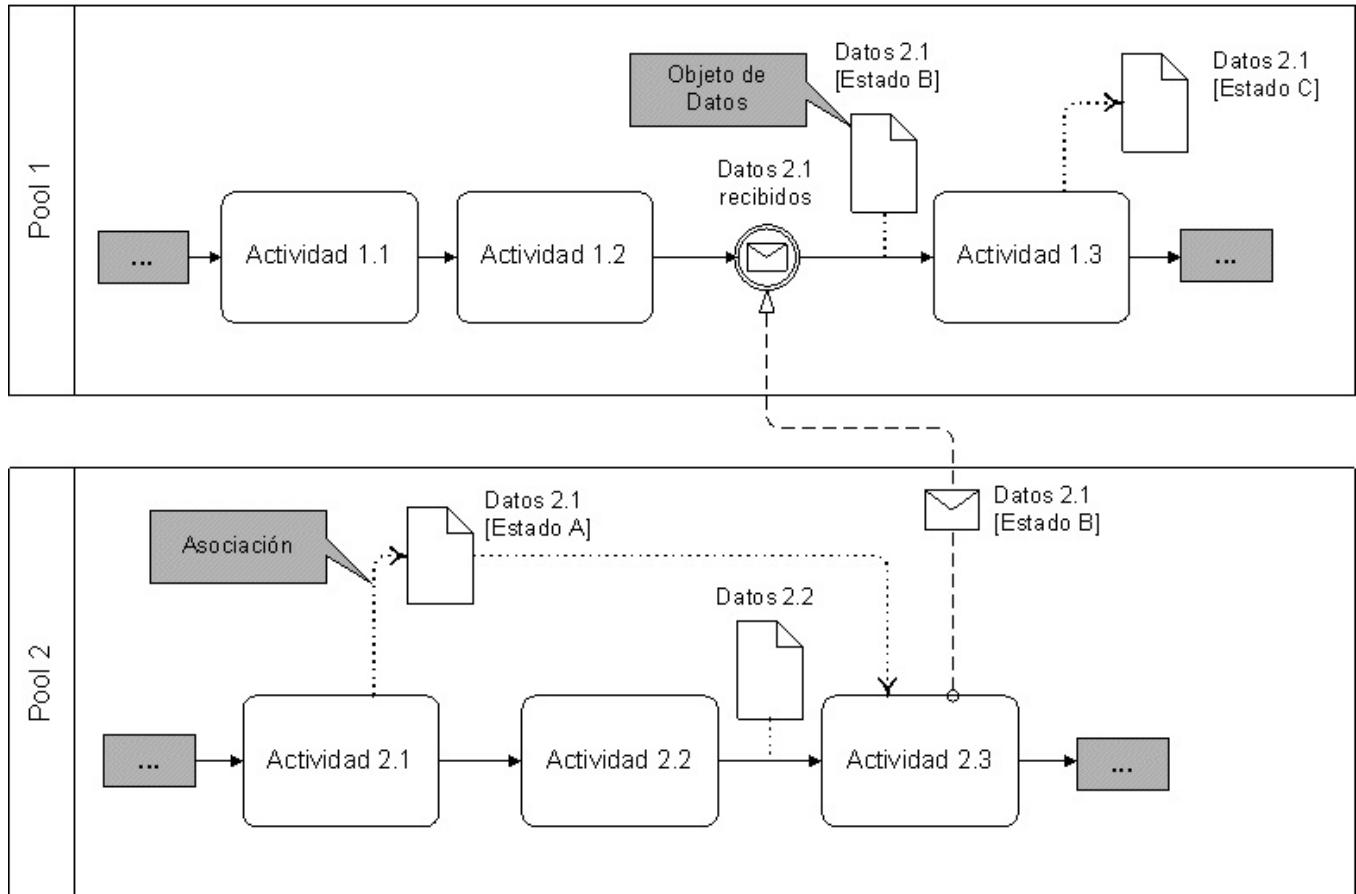


Figura 2.106: Ejemplo de uso con objetos de datos

El Pool 1 está esperando este mensaje y al recibirlo, envía el objeto de datos 2.1 (con «Estado B») a la actividad 1.3, en donde el objeto de datos es procesado y recibe el «Estado C».

Hay que tener cuidado de no confundir los flujos de mensajes con los elementos de asociación, porque se parecen mucho. Las principales diferencias son:

Los flujos de mensajes se representan por medio de líneas intercaladas y la flecha es cerrada. El comienzo de flujo de mensaje lo indica un círculo pequeño pegado a linea intercalada.

Los elementos de asociación dirigida se representan por medio de una línea interpunteada y la flecha está abierta.

En BPMN 2.0 se le da mayor importancia a los objetos de datos que en la versión anterior, razón por la cual reciben en la nueva versión una categoría propia. Esto se debe a que con BPMN 2.0 se persigue el objetivo de poder interpretar y ejecutar directamente los diagramas para implementarlos técnicamente y para cumplir este objetivo es necesario considerar los datos que controla la lógica de los procesos. En este contexto se introdujeron en BPMN 2.0 otros tipos de objetos de datos que ilustramos a modo de nuestro ejemplo de la pizzería en la figura 2.107.

Si a la pizzería entra una orden de pedido, podemos representarlo por medio de un objeto de datos que contiene una flecha tipo input (ubicada en la esquina superior izquierda del objeto). El pedido puede contener la entrega de una o más pizzas que es representado por la propiedad de múltiple como lo hemos visto ya en otros objetos de la notación. El objeto múltiple indica que se tendrán que hornear tantas pizzas como contiene el pedido. La actividad cobrar pizza necesita este objeto de datos como guía de despacho y que se pueda proceder a cobrar la entrega. La contabilidad va a registrar la venta en un sistema contable que es representado por un objeto tipo archivo. El objeto tipo output se representa con una flecha en negrita (ubicada en la esquina superior izquierda del objeto).

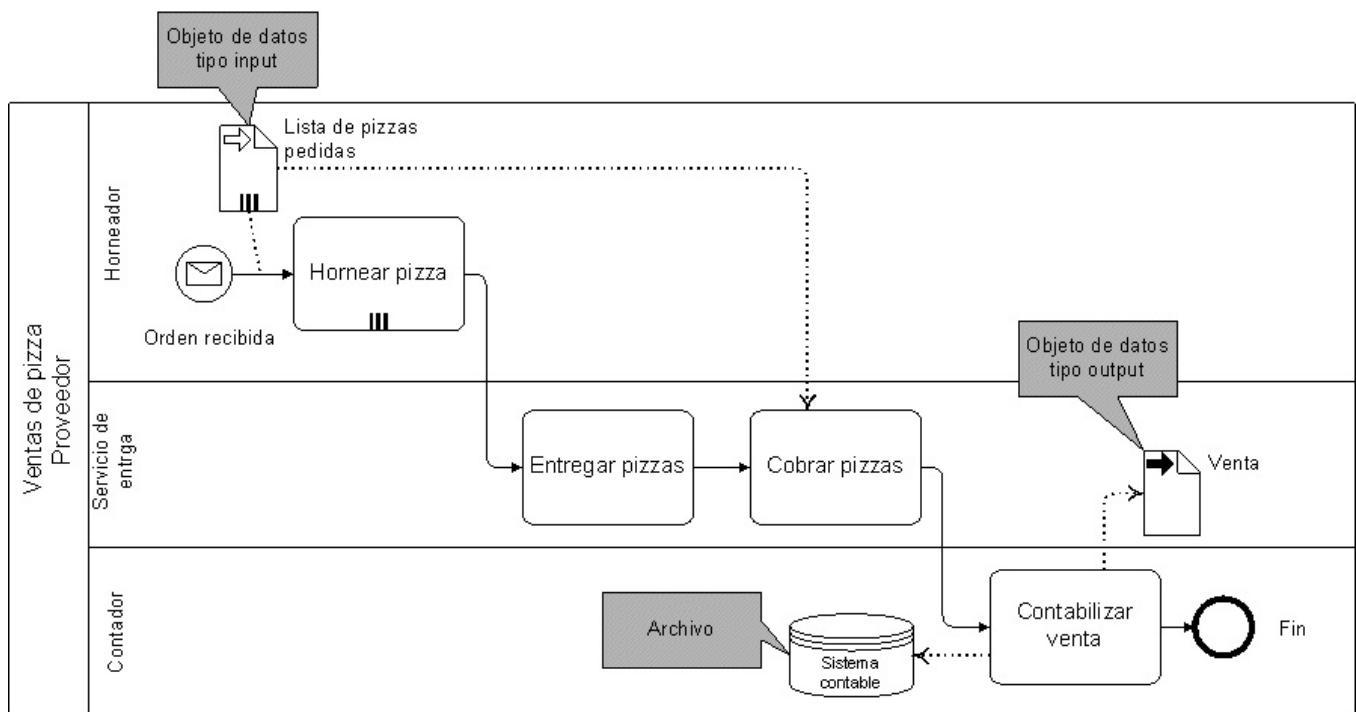


Figura 2.107: Nuevos símbolos de datos en BPMN 2.0

2.11. Artefactos

BPMN contiene también una categoría de elementos que sirven para una mejor explicación o visualización gráfica, pero que de ninguna forma tiene alguna influencia en la lógica de los procesos, por lo cual los «artefactos» no son interpretados por un motor de workflow.

2.11.1. Comentarios y agrupación

Existen dos tipos de artefactos que pueden servir para mejorar la interpretación y la legibilidad de los diagramas. El artefacto tipo «comentario» que se relaciona con los objetos de flujo por medio del elemento de asociación simple (línea interpunteada). En la figura 2.108 vemos el uso del artefacto tipo comentario para explicar donde se cocina y en que lugar se realizará la cena, pero también para revelar el origen de la

receta. En la figura 2.7 ocupamos el comentario para anotar los tiempos de ejecución de las actividades y en la figura 2.72 para especificar la condición de cancelación.

Para demarcar el lugar donde se cocina y donde se realizará la cena ocupamos el elemento tipo agrupación, pero habrá que tener cuidado de confundir este signo de agrupación con uno del tipo subprocesso abierto, que si tiene un significado en BPMN. El signo de agrupación lo puede ocupar libremente, incluso puede agrupar elementos que van más allá de los límites de uno o varios pools. El elemento de agrupación puede ser de gran utilidad para demarcar ciertos dominios en los procesos. Nosotros vamos a emplear a menudo este signo en los próximos capítulos, con la finalidad de explicar mejor nuestros ejemplos.

2.11. ARTEFACTOS 109

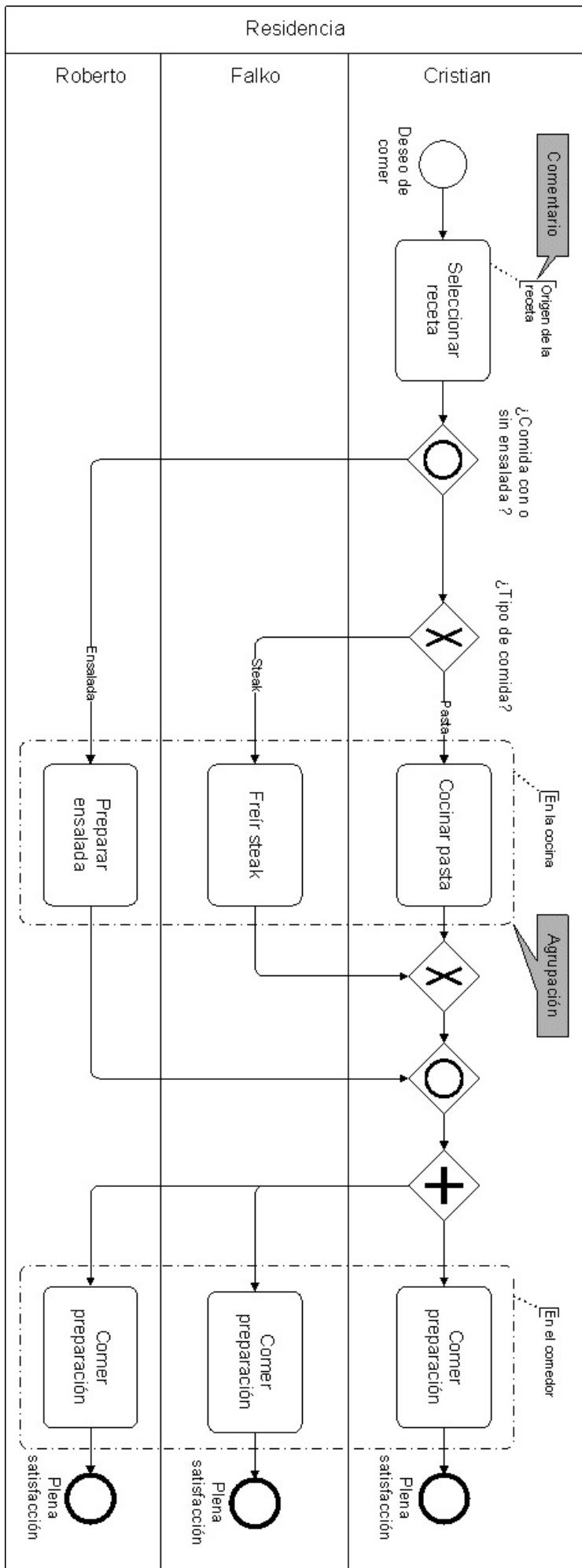


Figura 2.108: Caso con utilización de Comentarios y Agrupaciones

2.11.2. Artefactos y simbología propia

La notación de BPMN también permite introducir simbología propia como artefactos. Con esta posibilidad se pueden enriquecer los diagramas con artefactos ilustrativos típicos de un negocio. En la figura 2.109 introducimos en nuestro ejemplo una bicicleta para el servicio de entrega y un computador, para indicar los recursos que utilizan algunas actividades. Estos artefactos propios se relacionan igual que los demás, con el elemento de asociación a los objetos del modelo.

Nuestro Tip en BPMN

La posibilidad de usar artefactos propios depende de si las herramientas que está utilizando apoyan esta funcionalidad. Hasta el momento hemos visto pocas herramientas que permitan ampliar el metamodelo.

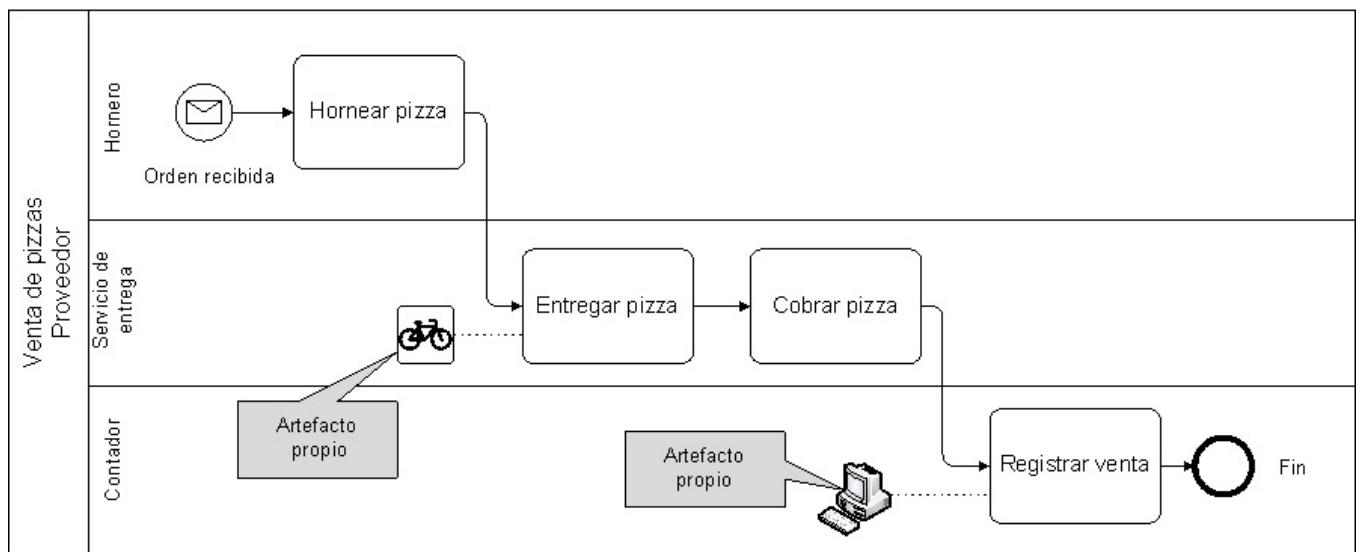


Figura 2.109: Artefactos propios

2.12. Modelos de Coreografía y Conversación

En la sección 2.9 vimos la importancia que se le da en BPMN a la interacción entre los diferentes participantes en un proceso de negocio. Con BPMN 2.0 se ofrecen dos nuevas formas de modelar esta interacción en forma más compacta y expresiva. Se trata de los diagramas de coreografía y de conversación que se concentran en forma exclusiva de representar el intercambio de información entre los participantes independientes. Debido a que la versión BPMN 2.0 es reciente, aún no podemos reportar la existencia de experiencias reales en el uso de estos nuevos tipo de diagramas. Debido a lo anterior, en esta versión del libro nos vamos a restringir a explicar con un ejemplo el potencial de uso que tienen estos nuevos modelos en BPMN.

En la figura 2.110 encontrará nuestro ejemplo de colaboración en el proceso de negocio del pedido de pizza que tratamos en la sección 2.9. El diagrama sólo lo ampliamos por los nuevos elementos de BPMN 2.0 de objetos de mensajería. En la figura 2.110 b) encontrará el mismo modelo diagramado con coreografía. Este tipo de diagrama reduce la lógica de los procesos a observar sólo el intercambio de información entre los participantes. También vimos este efecto en la sección 2.9.4, en donde se cerraron todos los pools del diagrama de colaboración, pero a diferencia del caso anterior en los diagramas de coreografía, podemos agregar lógica entre el intercambio de información de los participantes. Los diagramas de coreografía son mucho más compactos porque se concentran sólo en el intercambio de información entre los participantes y su debida lógica. El participante que envía información, lleva en el cuadro de coreografía un fondo «blanco» y el participante que recibe y procesa la información con un fondo «gris». No existe una regla en el objeto de coreografía con respecto al orden «arriba o abajo» del participante que envía o recibe; pueden posicionarse como mejor le parezca al modelador. En un diagrama de coreografía también pueden declararse los objetos como subprocessos. En este caso el subprocesso contiene su propia lógica de coreografía.

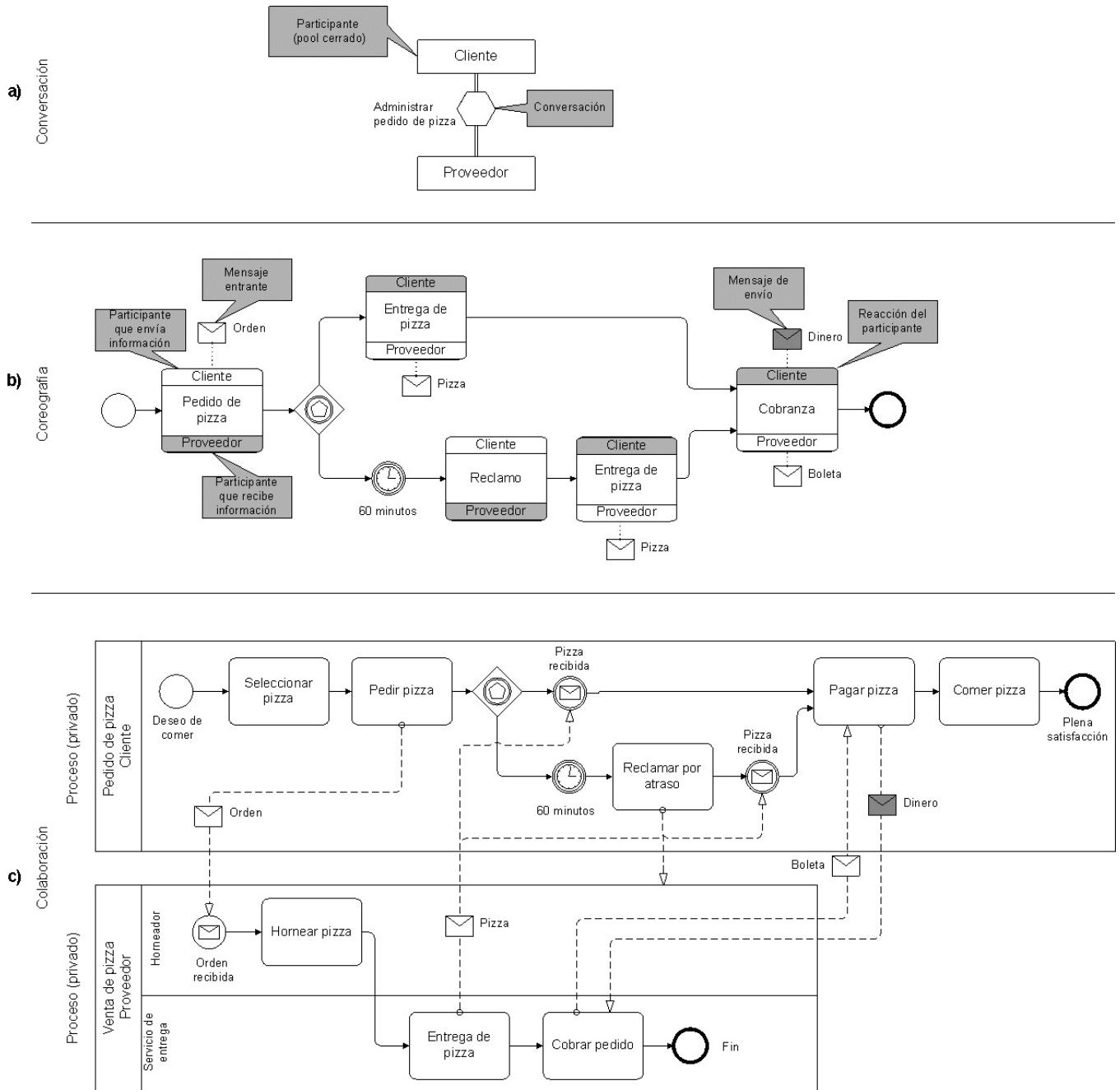


Figura 2.110: Proceso de Compra de Pizza como a) Modelo de Conversación b) Modelo de Coreografía y c) Modelo de Colaboración

En la figura 2.110 a), modelamos para el mismo caso en forma muy abstracta el correspondiente modelo de conversación. Estos diagramas representan la forma más compacta en BPMN para modelar el intercambio de información entre participantes independientes en BPMN. Para ser más correctos podríamos decir que en realidad los diagramas de conversación no son un «nuevo» tipo de diagrama en BPMN, sino que representa un modelo de colaboración cerrado, con la salvedad que el diagrama de conversación permite indicar cardinalidades en la relación entre pools.

Muchos analistas se preguntarán si tienen algún sentido estos diagramas más compactos. Algunos autores y expertos de BPMN, los encuentran innecesarios [Silver09]. ¡Nosotros no compartimos esta opinión! Tenemos por lo

menos dos situaciones muy importantes en la vida real, en donde los diagramas de coreografía pueden ser de gran utilidad en la coordinación de flujos complejos. Organizaciones grandes, globales que interoperan con muchas áreas, sub-empresas, filiales nacionales y/o internacionales, mantienen un flujo complejo y cambiante de información en sus procesos de negocio. Reducir el flujo de información y su debida lógica entre los participantes (coreografía), puede ser de gran utilidad para coordinar protocolos de intercambio de información entre las partes. El otro caso es el intercambio de información entre las empresas (B2B), o entre empresas y organizaciones públicas (B2G), o entre empresas y clientes (B2C). La tendencia de extender la integración de los procesos hacia afuera (clientes, proveedores, entes reguladores, gobierno, etc.) es cada vez mayor. Los diagramas de coreografía y/o de conversación son un medio ideal para acordar protocolos de intercambio de información, los cuales más tarde se transforman en servicios.

El grado de aplicabilidad de estos diagramas tienen mucho que ver la complejidad del entorno de observación. No tiene mucho sentido aplicar esta técnica por ejemplo para una empresa Pyme (Pequeña y mediana empresa) que hace muchos años tiene una relación acordada con un proveedor, ni tampoco tendrá mucho sentido que una sociedad de tres personas que se coordinan perfectamente con reuniones semanales, se pongan a modelar sus procesos de colaboración, pero ¿qué opina el lector del caso de una organización global con más de 50 mil empleados, con muchas líneas de negocio, más de 1.500 proveedores y sucursales repartidas por todo el mundo?

En un proyecto de investigación de logística se hizo la prueba de modelar una lógica de coreografía que trata de la integración de un proceso de compras entre un retailer y los proveedores a través de un portal B2B.

Integración Proceso Compras: Retailer-Proveedor a través de Portal B2B

Diagrama de Coreografía BPMN 2.0

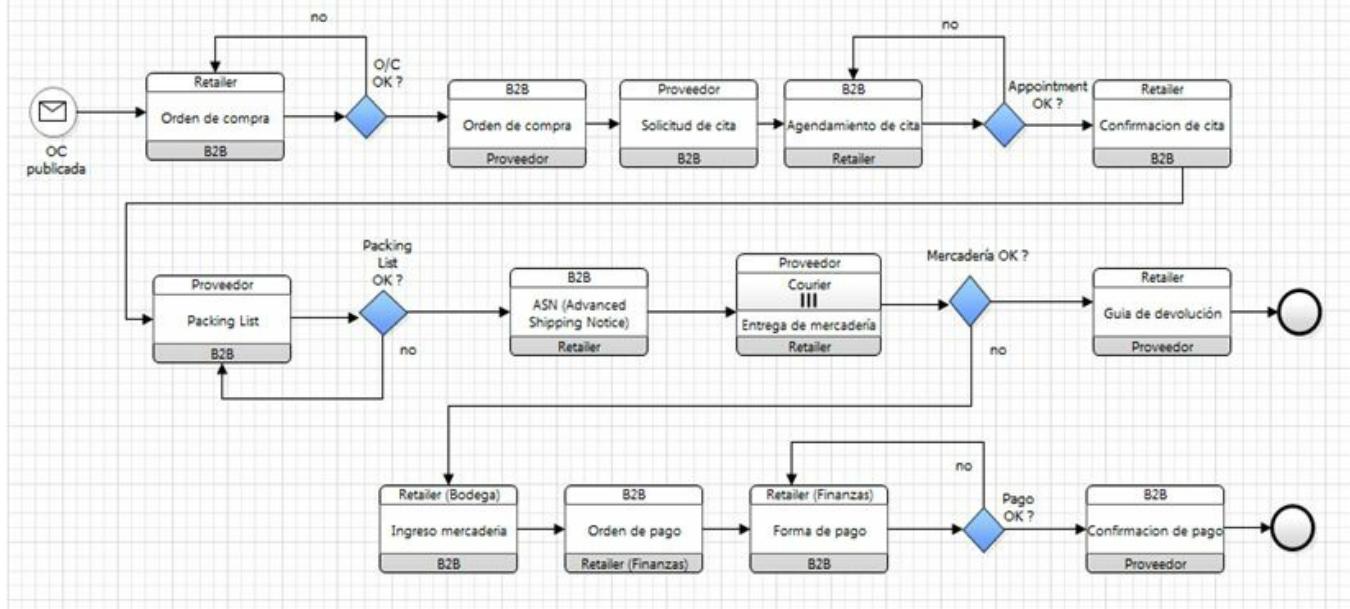


Figura 2.111: Modelo de Coreografía en BPMN2.0 para el Proceso de Compra y Recepción de Mercadería vía B2B (Inbound Logistics)

Fuente: [HitpassLeiva12]

En la Figura 2.111 se puede apreciar el resultado modelado con el nuevo diagrama de coreografía BPMN 2.0, para el proceso de Compras y Recepción, de un Retailer y un Proveedor, utilizando como herramienta de integración y coreografía al Portal B2B. Este portal web permite la sincronización de procesos en un ambiente colaborativo y favorece la coreografía de los distintos procesos, tales como:

Generación y Publicación de Órdenes de Compra, por parte del retailer (upload)
 Recuperar (download) y procesar Órdenes de Compra, por parte del proveedor
 Agendamiento de Citas de entrega de mercadería, por parte del proveedor (Appointments)

Generación y Publicación del Packing List (ASN), por parte del proveedor Entrega mercadería física del proveedor, y proceso de recepción y devolución, del retailer

Proceso de pago al proveedor, por parte del retailer

En general, una de las grandes ventajas de estos tipos de diagramas de coreografía es que se puede representar en forma mucho más compacta la interacción entre los participantes.

2.13. BPMN comparado con otras Notaciones

Muchos lectores que se interesan por BPMN conocen algunas otras notaciones para modelar procesos. Seguramente también se preguntarán si tiene sentido cambiarse a

BPMN y qué aspectos hay que considerar en esta nueva técnica de modelamiento. Según la región donde esté radicado el lector y la escuela por la que ha pasado, habrá conocido o aplicado diferentes notaciones. En algunas regiones nos encontramos con una mayor influencia del pensamiento europeo y en otras con una mayor influencia de la escuela norteamericana, razón por la cual se ha escogido para una comparación, aquellas técnicas más representativas de ambas regiones (Nota: El autor de la versión hispana de este libro ha reemplazado algunas notaciones propias de países de habla alemana, por otras que son más típicas en la región hispana).

Cuando se empezó a desarrollar BPMN se revisaron muchas otras notaciones de modelamiento. Los miembros de la OMG aportaron con sus conocimientos y experiencias con muchas notaciones existentes, de las cuales algunas de ellas influyeron en el desarrollo del estándar, como por ejemplo UML Activity Diagram, UML EDOC Business Processes, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, and Event-Process Chains (EPCs)[Obj11]. Algunas de estas notaciones que se revisaron son de carácter técnico e influyeron en el desarrollo del concepto de colaboración y coreografía (UML EDOC, ebXML BPSS, RosettaNet).

ebXML BPSS (Electronic business using XML), UML EDOC (Profile for Enterprise Distributed Object Computing (EDOC)), RosettaNet (Metodología estándar para supply chain), LOVeM (Técnica para modelar interfaces de IBM).

Otras notaciones más orientadas al negocio se revisaron para extraer ideas en la parte conceptual del modelamiento de procesos de negocio, como EPC, IDEF y UML Activity Diagram. Con el fin de entender la evolución que han tenido en las últimas cuatro décadas las técnicas de modelamiento, le presentamos al lector en la siguiente sección una breve reseña histórica de estas técnicas y notaciones, sobre todo enfocado a explicar los supuestos cambios de paradigmas. Con el objetivo de no tentarse en comparar peras con manzanas, hemos incluido una pequeña sección 2.13.2 en la cual se muestran criterios de clasificación de técnicas de modelamiento. Finalmente comparamos con BPMN las dos notaciones más difundidas actualmente para modelar procesos de negocio (Secciones 2.13.3 y 2.13.4).

La mayor debilidad de las notaciones comparadas con BPMN, es la insuficiencia estructural para modelar la lógica entre los participantes autónomos de los procesos, es decir de modelar la colaboración entre los procesos. Este aspecto conceptual de colaboración y coreografía se convierte hoy en día en un factor crítico, si pensamos que los desafíos actuales tienden a un grado cada vez mayor de integración de los procesos en una organización y sobre todo con sus agentes externos (proveedores, clientes, entes gubernamentales, entes reguladores, etc.).

2.13.1. Historia de técnicas para modelado de sistemas y procesos

Durante la historia se han ido desarrollado varias técnicas de modelado, cada una con énfasis y orientaciones distintas. A partir de los años 60 se empezaron a desarrollar técnicas de modelado sobre todo orientadas al desarrollo de sistemas y la mayoría de éstas, enfocadas al modelamiento de datos y funciones. Lo que se buscaba encontrar eran vistas de datos normalizadas que debían ser administradas por la funcionalidad que se requería para el negocio, razón por la cual dominaban las técnicas centradas en flujos de datos, como lo fue el análisis estructurado (en inglés: Structured Analysis). El método de análisis estructurado se convirtió en su época en sinónimo del análisis de flujo de datos. Se desarrollaron herramientas para documentar y administrar los flujos de datos. Las herramientas eran esenciales para documentar los sistemas existentes y para determinar los requerimientos de información por medio del método estructurado. Los analistas deseaban conocer las respuestas a cuatro preguntas específicas: ¿Qué funciones integran el sistema? ¿Qué datos necesita cada función? ¿Qué datos deben ser almacenados? ¿Qué datos ingresan y abandonan el sistema? De lo dicho anterior queda claro que se daba gran importancia a los datos. Para el desarrollo de aplicaciones de negocios fue una técnica apropiada, la lógica de negocio en los sistemas de información está centrada en los datos, en el «qué», pero desde el punto de vista de los procesos, la lógica de negocio se centra en el «cómo». En aquella época, en que dominaba el pensamiento de administrar la información del negocio, el desafío era desarrollar sistemas integrados con bases de datos normalizadas, es decir libre de redundancia en la mantención de los datos.

Los métodos para el análisis de flujo de datos fueron desarrollados por dos organizaciones: Yourdon Inc., una empresa consultora que promovió y publicó los famosos libros de DeMarco, Weinberg y Page-Jones. McDonnell-Douglas, con el trabajo y publicaciones de Gane y Sarson, también influyeron en la popularidad del análisis estructurado[Senn92].

Los diagramas lógicos de flujo de datos se pueden diagramar con sólo cuatro objetos sencillos. El uso de símbolos para cada objeto depende de qué enfoque se utilice, el de Yourdon o el de Gane y Sarson (ver figura 2.112):

Notaciones Alternativas

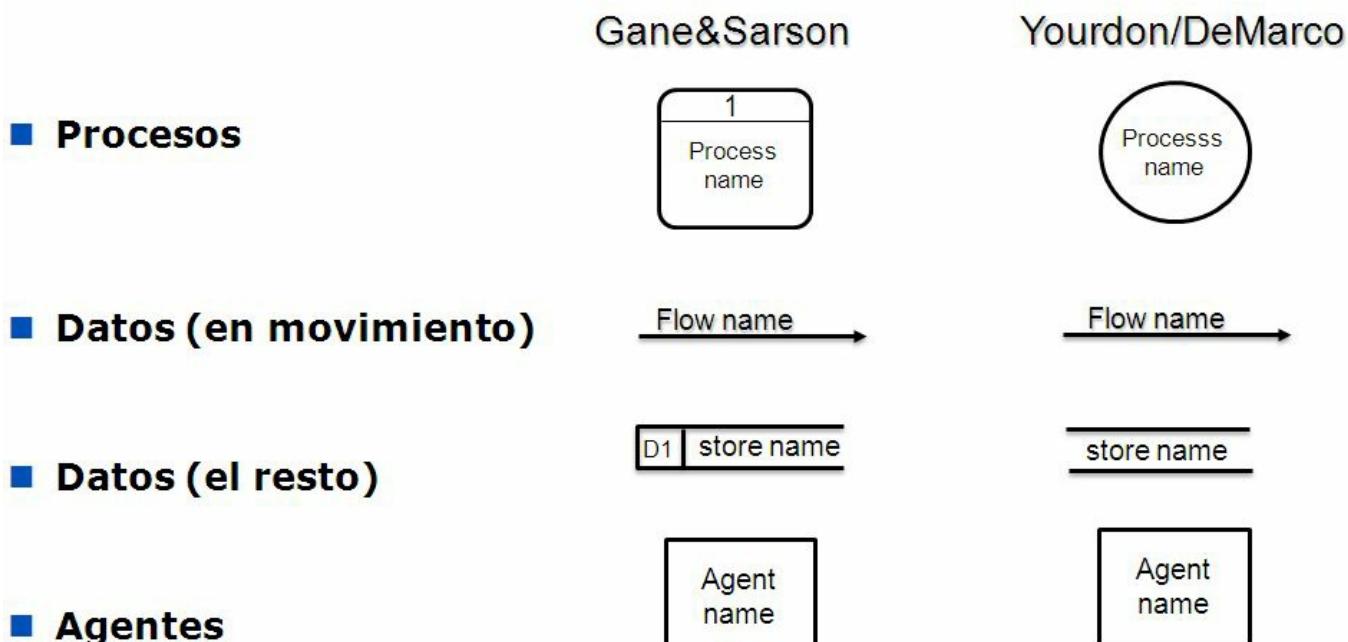


Figura 2.112: Objetos del Diagrama Análisis Estructurado según Yourdon o Gane y Sarson

Cada objeto en un diagrama tiene una etiqueta con un nombre descriptivo. Los nombres de las funciones también reciben un número identificador. El número asignado a una determinada función no tiene relación con la secuencia en un proceso (flujo de control). Sirve estrictamente para la identificación y tiene un valor adicional cuando se estudian los componentes que integran un sistema. La figura 2.113 muestra en forma genérica la descomposición jerárquica de flujos de datos y de funciones de un diagrama de análisis estructurado y la figura 2.114 un ejemplo del primer nivel de diagrama del contexto.

Nota: En el análisis estructurado a las funciones también se les llama «procesos» porque procesan datos, pero el lector no debe confundirlos con la taxonomía que usa hoy en día para la definición de procesos.

Diagrama de Contexto

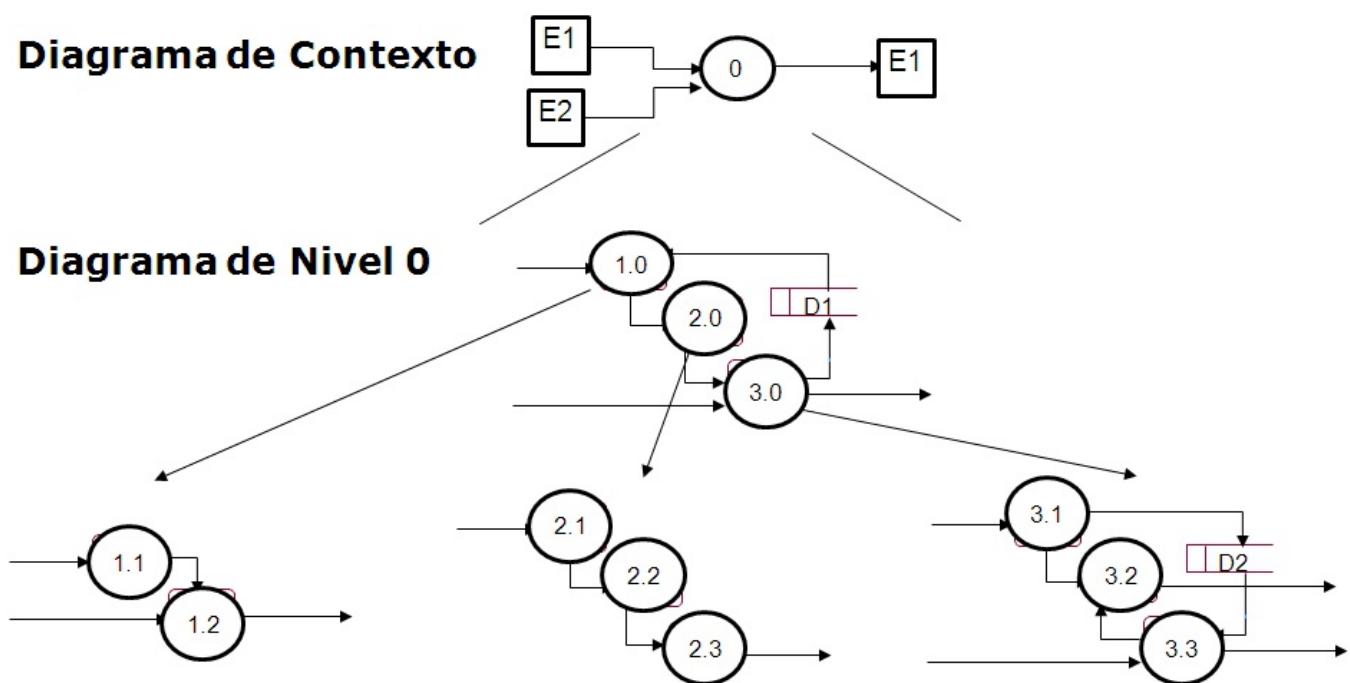


Figura 2.113: Descomposición jerárquica de un diagrama de análisis estructurado

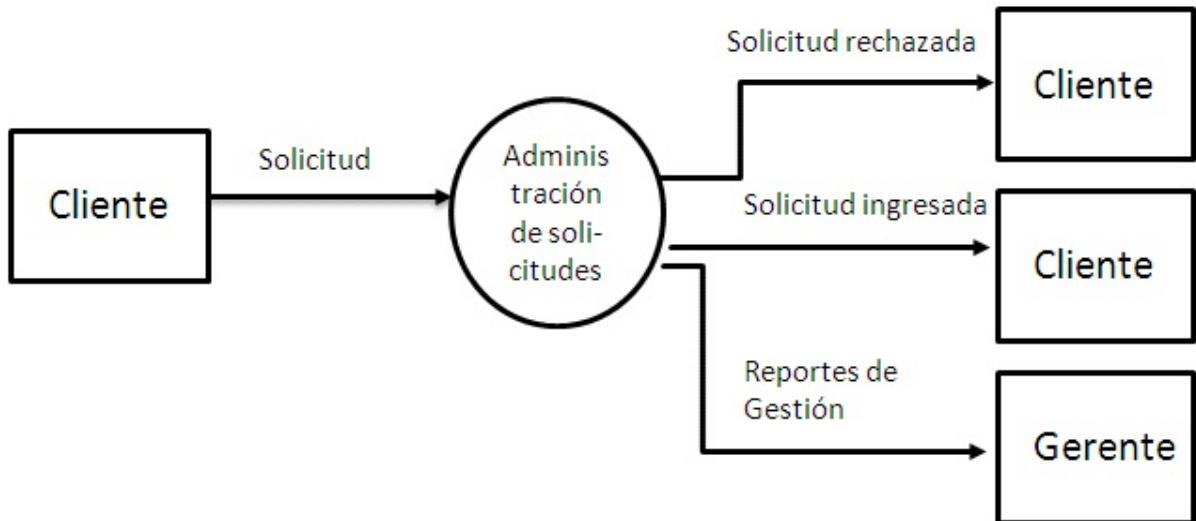


Figura 2.114: Ejemplo Diagrama del Contexto

Hoy en día ¿se recomienda utilizar técnicas de análisis estructurado para modelar procesos? Para responder a esta pregunta tenemos que primero entender el pensamiento y el desafío de hace dos o tres décadas atrás y el de hoy. En aquella época el desafío de las organizaciones era lograr administrar la lógica de la información del negocio por medio de sistemas informáticos, centrados en el «qué», es decir en los datos. Aún no se planteaba, por lo menos desde el punto de vista de la automatización, el «cómo» trabajar mejor o más eficiente. Para entender mejor el pensamiento de aquella época vamos a citar a Senn en su libro de análisis

estructurado de principios de los años 90 [Senn92]:

«Los datos son la guía de las actividades de la empresa. Ellos pueden iniciar eventos (por ejemplo, los datos sobre nuevos pedidos) y ser procesados para dar información útil al personal que desea saber qué tan bien se han manejado los eventos (al medir la calidad y tasa de trabajo, rentabilidad, etc.). El análisis de sistemas conoce el papel central que tienen los datos de la empresa en las organizaciones. Seguir el flujo de los datos por todos los procesos de la empresa, que es la finalidad del análisis de flujo de datos, les dice mucho a los analistas cómo se alcanzan los objetivos de la organización. El análisis de flujo de datos estudia el empleo de los datos en cada actividad.»

Hoy en día diríamos, los procesos son la guía de las actividades de la empresa, ¡no los datos! Con esta pequeña reseña histórica, el lector se habrá dado cuenta del cambio de paradigma. ¿Pero por qué este cambio? ¿Se habrán equivocado antes, o estaremos equivocados hoy en día? La primera respuesta es que la pregunta está mal planteada, no se equivocaron antes ni tampoco hoy en día. Desde el punto de vista del desarrollo de sistema de información, los datos y funciones que administran los datos siguen teniendo la misma importancia que antes, sólo que hoy en día se parte de la base que los sistemas de información existen para casi todos los rubros, sea que las organizaciones hayan adquirido software estándar o hayan desarrollado soluciones propias. Es la evolución, sobre todo técnica, que ha permitido ahora poder preocuparnos de «cómo» poder organizar mejor la forma en que trabajamos y es justamente esta la pregunta a la que quiere responder la nueva disciplina llamada «Gestión por Procesos de Negocio (en inglés: Business Process Management, BPM)». Para los procesos no es el flujo de datos lo esencial, sino el flujo de control. Entonces no nos encontramos ante un cambio de paradigma, sino más bien en un nivel de evolución mayor. En el desarrollo de sistemas de información, los datos siguen siendo tan importantes como antes, aunque tenemos que admitir que el enfoque de análisis estructurado fue reemplazado a mediados de los noventa por el enfoque orientado al objeto (Object Oriented Analysis (OOA), Object Oriented Design (OOD) y Object Oriented Programming (OOP)). Las metodologías orientadas al objeto se materializaron principalmente en la familia de modelos Unified Modeling Language (UML) que presentaremos en relación a modelado de procesos en la sección 2.13.2.

Al introducir esta sección se explicó que durante la historia se han ido desarrollando varias técnicas de modelado, cada una con énfasis y orientaciones distintas. En las décadas de los 60 hasta los 90 predominaban las técnicas de modelado orientadas al desarrollo de sistemas y a partir de los 80 aparecen las primeras técnicas orientadas al modelado de procesos, la más difundida fue la técnica del «Event driven Process Chains (EPC)» que presentaremos en la sección 2.13.3, pero también se incluyeron o ampliaron técnicas de flujo de control (modelado de procesos) a las llamadas familias de técnicas de modelado como IDEF y UML (ver sección 2.13.2).

IDEF (Integrated Definition for Function Modeling) es una familia de técnicas de modelamiento, que ofrece una serie de modelos para modelar datos, funciones, procesos, recursos, simulación y otros. Sus inicios se remontan a principios de los años 70 desarrollado para la Fuerza Aérea Estadounidense bajo el nombre inicial de ICAM (Integrated Computer Aided Manufacturing)[IDEF10].

La familia IDEF, consiste en un gran número de técnicas, entre las cuales se destaca IDEF0 e IDEF3, que son aquellas relacionadas con los procesos de negocio, aunque existen otras versiones como IDEF1(Modelo de Información), IDEF1X (Modelamiento de Datos), IDEF2 (Simulación), IDEF4 (Diseño orientado a objetos), IDEF5 (Descripción Ontológica), IDEF9 (Descubrimiento de restricciones de negocios) y otros. La técnica IDEF0, está diseñada para modelar las decisiones, acciones y actividades de una organización u otro sistema, y representa la perspectiva funcional de modelado. Permite describir los procesos y sus interfaces así como elaborar la documentación que permita el seguimiento en sus etapas de desarrollo. IDEF0 utiliza sólo un tipo de anotación en sus representaciones gráficas conocido como ICOM (Input-Control-Output-Mechanism), que fue deducida de los diagramas de análisis estructurado. La representación estática de sus diagramas no permite visualizar el modelado de las perspectivas de comportamiento o de información. Para vencer dichas limitaciones, se desarrolló IDEF3 (Process Description Capture), que describe a los procesos como secuencias ordenadas de hechos o actividades, representando el cómo, y mostrando la visión dinámica de los procesos.

2.13.2. Clasificación de las técnicas de modelamiento

Como el lector se habrá dado cuenta en nuestra breve reseña histórica, a lo largo del tiempo se han desarrollado muchas técnicas de modelado, de las cuales sólo presentamos algunas, las más conocidas. Cuando el lector se encuentre con una de ellas será importante saber clasificarlas, de lo contrario será difícil de evaluarlas para un fin determinado o para compararlas con otras notaciones.

Formalmente podemos hacer la primera gran división en metodologías basadas en técnicas de lenguaje estructurado (script) y metodologías basadas en técnicas de diagramación. Las notaciones del tipo de script están muy cercas de los lenguajes de programación (Ejemplos: BPEL o lenguajes propietarios de herramientas de software), son formalmente precisas, pero su expresividad visual es prácticamente nula, por lo que sería muy difícil de emplearlas con usuarios de negocio.

Las metodologías basadas en técnicas de diagramación, las podemos clasificar en técnicas orientadas al flujo de datos, al flujo de control y orientadas al objeto[Gad10](ver figura 2.115):

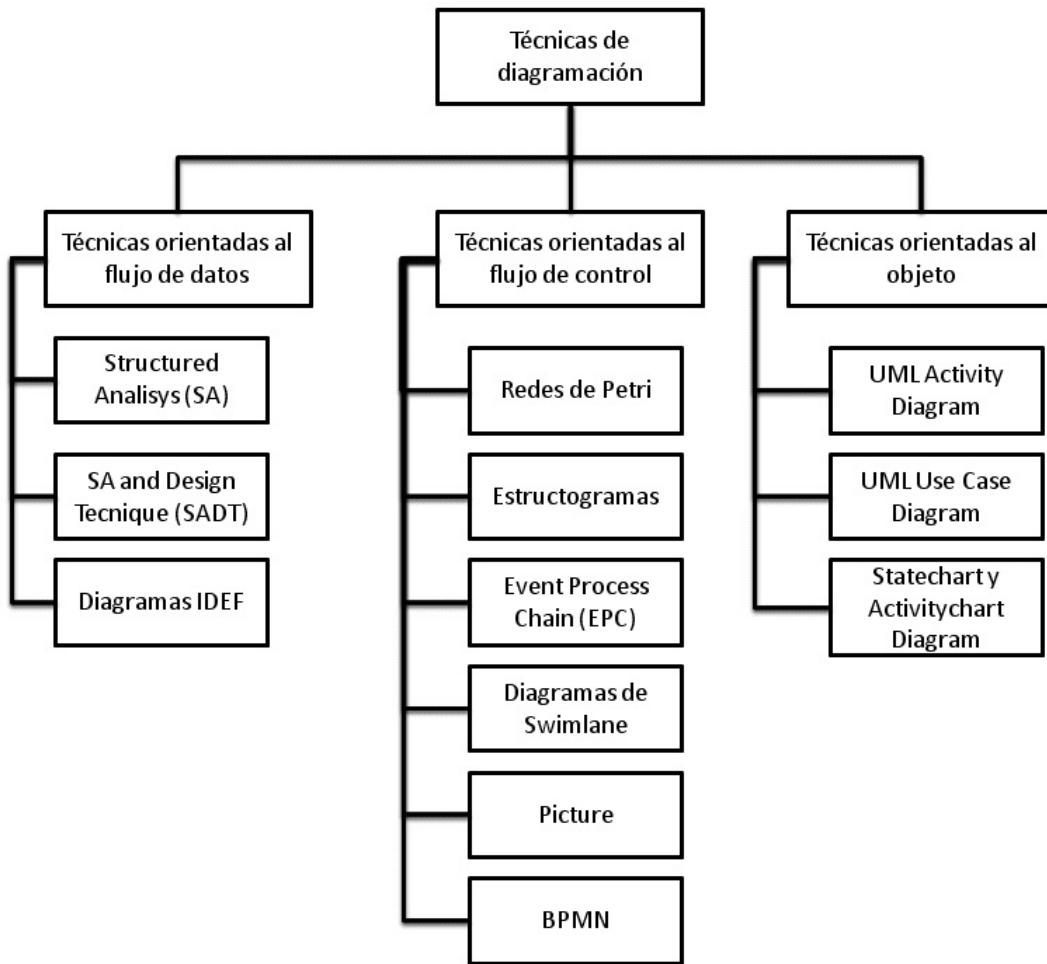


Figura 2.115: Clasificación de algunas técnicas de diagramación para modelamiento de procesos

El uso de técnicas orientadas al flujo de datos se están empleando cada vez menos, de hecho actualmente para la especificación de desarrollo de sistemas se utilizan los diferentes diagramas de UML.

Una Red de Petri está formada por lugares (círculos), transiciones (rectángulo) y arcos dirigidos (flechas), así como por fichas que ocupan posiciones. Los lugares representan estados en los procesos (documentos, datos, recursos,etc.) y las transiciones la transformación de información (funciones, procesos, actividades, etc.). Los arcos relacionan lugares y transiciones. A diferencia de los modelos EPC, las redes de Petri no utilizan nodos de conexión (decisión), las conexiones se entienden como implícitas. Las redes de Petri se utilizan bastante en algunas organizaciones para modelar sistemas de workflow y procesos.

Nota: Las redes de Petri fueron definidas en los años 60 por Carl Adam Petri. Son una generalización de la teoría de autómatas que permite expresar eventos concurrentes.

Los estructogramas (debido a la estructura de bloques recibió esta técnica el nombre

de estructogramas) muestran el control de un proceso por medio de la descomposición de bloques estructurados, que se van descomponiendo cada vez más, hasta llegar al nivel de actividad. Los bloques van disminuyendo de tamaño dentro del mismo diagrama a medida que se van descomponiendo los procesos. Los estructogramas son isomorfos con los diagramas de flujo. Todo lo que se puede representar con este tipo de diagramas se puede representar con un diagrama de flujo de control. Las únicas excepciones se dan en las instrucciones «*goto*, *break* y *continue*». La técnica de diagramación de estructogramas (técnica de diagramas en bloque para la programación estructurada desarrollada en los años 70 por Nassi-Scheinerman) que fue heredado de la ingeniería de software, no logró mayor difusión, razón por la cual no la seguiremos tratando.

Gran difusión logró la notación EPC, razón por la cual le dedicaremos una sección propia a esta técnica que podemos clasificarla dentro del grupo de las técnicas orientadas al flujo de control.

Los diagramas de swimlane fueron desarrollados a principios de los 90 por H.F.Binner[Gad10] bajo el nombre «RPO: Representación de Procesos Organizacionales (en alemán ODP: Organisationsprozessdarstellung). El nombre «swimlane» se utiliza en analogía a los carriles para nadar, en los cuales los actores de los procesos que representan una unidad organizacional o un rol, intercambian información a través de un flujo de secuencia (control) durante la ejecución de los pasos de un proceso hasta que llegue a término. Parte de esta notación fue adoptada en muchas otras técnicas de modelamiento, así también en BPMN.

Los diagramas del tipo Picture utilizan dispositivos con imágenes de objetos reales, para representar tipos de actividades en procesos. Esta técnica fue desarrollada en el año 2007 por el «European Research Center for Information Systems»[Gad10] con el propósito de diagramar mapas de procesos para la administración pública. Existen alrededor de 24 dispositivos de imágenes y otros elementos para modelar procesos. A diferencia de otras notaciones, la cantidad de tipos de objetos se reduce notablemente, también debido a que muchos elementos descriptivos se integran a través de atributos a los dispositivos. Sin embargo, esta técnica no alcanza un grado de formalización que sea suficiente para modelar procesos a nivel operacional.

Antes que apareciera BPMN se difundieron bastante algunas técnicas orientadas al objeto para modelar procesos, sobre todo UML Activity Diagram (diagramas de los procesos a nivel descriptivo) y diagramas UML Use Case (Casos de uso en un nivel más detallado que describen el flujo entre actividades y unidades organizacionales). La técnica de UML Activity Diagram la trataremos en comparación a BPMN en la sección 2.13.4.

La técnica de Statechart Diagram fue desarrollada por Harel en los años 90[Gad10] como propuesta para modelar relaciones complejas en los flujos de los procesos. El

flujo de control entre las actividades de un proceso se describe con ayuda de arcos dirigidos. El correspondiente diagrama Activitychart describe el flujo de datos. El flujo de control se define entre estados y actividades. Statecharts reaccionan a reglas del tipo Event-Condition-Action-Rules (ECA). La especialidad de esta técnica es el modelamiento de flujo de control considerando el flujo de datos.

En una encuesta realizada en el año 2007 por Gadatsch[Gad10] a grandes organizaciones se pudo constatar, que la técnica EPC, la de Swimlane Diagram y UML, eran relativamente las más utilizadas para el modelamiento de procesos. El promedio de utilización (respuestas múltiples estaban permitidas) lo encuentra el lector en la siguiente enumeración:

EPC (43,1 %)
Swimlane (38,8 %)
UML (21,6 %)
BPMN (16,4 %)
Redes de Petri (3,4 %)
SADT (44,0 %)
IDEF (2,6 %)

Encuestas más recientes (ver figura 0.1 en el prólogo de este libro) muestran un rápido o mejor dicho explosivo crecimiento hacia BPMN y junto a ello una disminución de las otras técnicas de modelamiento. La razón principal es porque BPMN se convirtió en un estándar oficial de la industria para modelar procesos que además ahora en la nueva versión contiene un metamodelo para intercambiar modelos entre herramientas y para implementar directamente estos con TI. Otra de las razones no menos importante es que es apoyado por casi todos los grandes fabricantes y proveedores de tecnología a nivel mundial. También en el ámbito científico y académico en BPMN se están desarrollando cada día más proyectos de investigación y conferencias.

2.13.3. Cadenas de procesos impulsadas por eventos (eEPC)

La notación de «Event driven Process Chain» (EPC) fue desarrollada en los años '80 por Keller, Nüttgens y Scheer en la Universidad de Saarland, Alemania y desde los años 90 se ha convertido en un estándar industrial en los países desarrollados, sobre todo en los de habla alemana. El EPC tiene su origen en las redes de Petri que define que todos los procesos se pueden explicar a través de los elementos « Estado - Transición -nuevo Estado ». El Prof. August Wilhem Scheer fundó una empresa llamada IDS-Scheer que desarrolló una herramienta mundialmente conocida como ARIS (Architecture of Integrated Systems) y en la cual integró en su vista principal «la vista de control» la notación para modelado de procesos EPC. En el año 1992 la empresa IDS-Scheer hizo una cooperación con SAP AG para documentar e integrar

los procesos de negocio de la solución de ERP de SAP. La integración de ARIS con SAP fue una de las principales razones que ARIS y EPC se hicieran mundialmente conocidas, más allá de los países de habla alemana. Hasta el año 2008 EPC fue la notación más difundida para el modelamiento de procesos, sobre todo en los países de habla alemana, pero con la aparición del estándar BPMN la situación cambió. Hasta el 2010, podemos constatar consultando varios medios, que el interés por BPMN ha sobrepasado al de EPC. Muchos analistas que emplearon EPC por muchos años se están preparando para cambiarse a la nueva notación, lo que no a todos los expertos de EPC les resulta fácil debido al distinto enfoque que tienen ambas notaciones. También IDS-Scheer reconoció la importancia del estándar e incluyó rápidamente la notación a su plataforma de ARIS (Nota: La empresa IDS Scheer fue adquirida en el año 2010 por Software AG).

Un proceso de negocio puede modelarse (a nivel funcional) con la estructura y diagrama del modelo EPC, que se componen de los elementos, eventos, funciones (actividades en BPMN), conectores (nodos de decisión) y unidades de la organización. Los conectores en EPC, (XOR-OR-AND) tienen casi el mismo significado que los Gateways en BPMN, con la diferencia que EPC no hace diferencia entre compuertas basadas en datos o eventos como en BPMN. En la versión extendida de EPC, conocida como eEPC (extended EPC), pueden utilizarse elementos que enriquecen el diagrama con mayor información como datos, aplicaciones, data cluster, documentos, materiales y objetivos. EPC permite al igual que BPMN la agregación de funciones como subprocessos o la conexión de diagramas a través de links.

Existen reglas para convertir procesos modelados en EPC a BPMN. Este ejercicio lo hicimos a modo de ejemplo que usted puede apreciar en la figura 2.116. Algo de cuidado tiene que tener al mapear los eventos a BPMN, porque desde el punto de vista estructural EPC no hace ninguna clasificación con respecto a los diferentes tipos de eventos. EPC por ejemplo, no conoce la diferencia entre eventos de inicio, intermedios, de término y tampoco hace la diferencia entre eventos de tiempo y de mensaje. Tampoco se puede utilizar en EPC el concepto de eventos sobrepuertos que permiten modelar situaciones como interrupciones, tratamiento de errores o de escalación.

Debido a su historia, la notación EPC aún tiene una gran difusión a nivel global y en general podemos concluir que el modelo EPC es simple y adecuado a la terminología del usuario final, pero inadecuado para llevarlo a un diseño de implementación técnica, razón por la cual no recomendamos seguir usándola a partir del nivel operacional.

2.13.4. UML Diagrama de actividades

El diagrama de actividades de UML (Unified Modeling Language) pertenece a una familia de 13 tipos de diagramas que existen en la versión actual 2.0. UML es administrado al igual que BPMN por la OMG (Object Management Group) como estándar, con la diferencia que UML es bastante más antiguo que BPMN (desde el año 1997). Sin embargo no hay que mal interpretar que BPMN reemplazaría a UML, porque UML fue desarrollado como técnica para el desarrollo de sistemas, mientras que BPMN para el modelamiento de procesos. A pesar de esta diferencia, se ha empleado mucho y sobre todo antes que apareciera BPMN, la técnica de diagramas de actividades de UML para modelar procesos. Se ha empleado preferentemente como técnica de especificación para el desarrollo del workflow embebido en sistemas de TI.

La notación de diagrama de actividades de UML es más amplia que la de EPC. Existen en UML objetos muy técnicos que ni siquiera se conocen en BPMN, sobre todo en el ámbito de especificación de parámetros y reglas para el procesamiento de objetos. En los aspectos relevantes para el modelamiento de procesos pueden traspasarse casi todos los elementos del diagrama de actividades de UML a BPMN, con la salvedad de las áreas de interrupción en UML que pueden modelarse por encima de los roles (o lanes en BPMN). Este caso no se puede traspasar uno a uno a BPMN, por ejemplo como subprocesso, sino que en BPMN habría que modelarlo como subprocesso global para que impacte sobre cada uno de los lanes afectados en BPMN (ver figura 2.117).

Para efectos de una especificación técnica de flujos en el desarrollo de sistemas, los diagramas de actividades de UML, especificando casos de uso, van a seguir siendo importantes, pero para la implementación de procesos de negocio BPMN es el estándar declarado y su notación es muy superior a la de UML. La definición técnica de procesos que van a ser interpretados directamente por un motor de workflow es el dominio de BPMN y no existe ninguna otra notación hasta el momento que supere en grado de expresividad a este nuevo estándar.

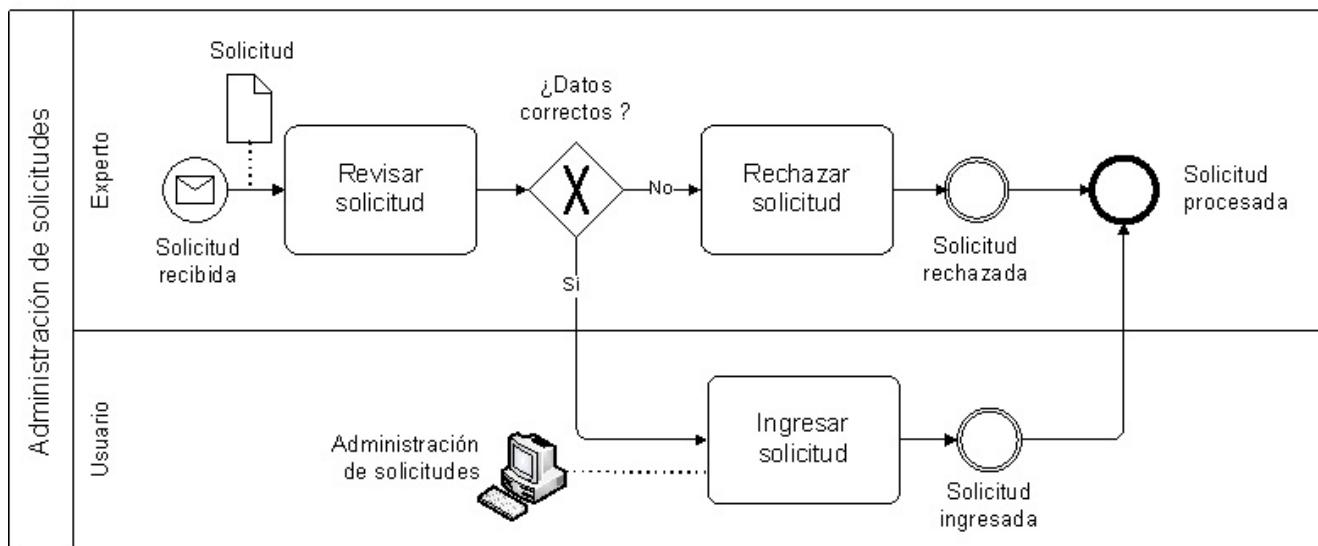
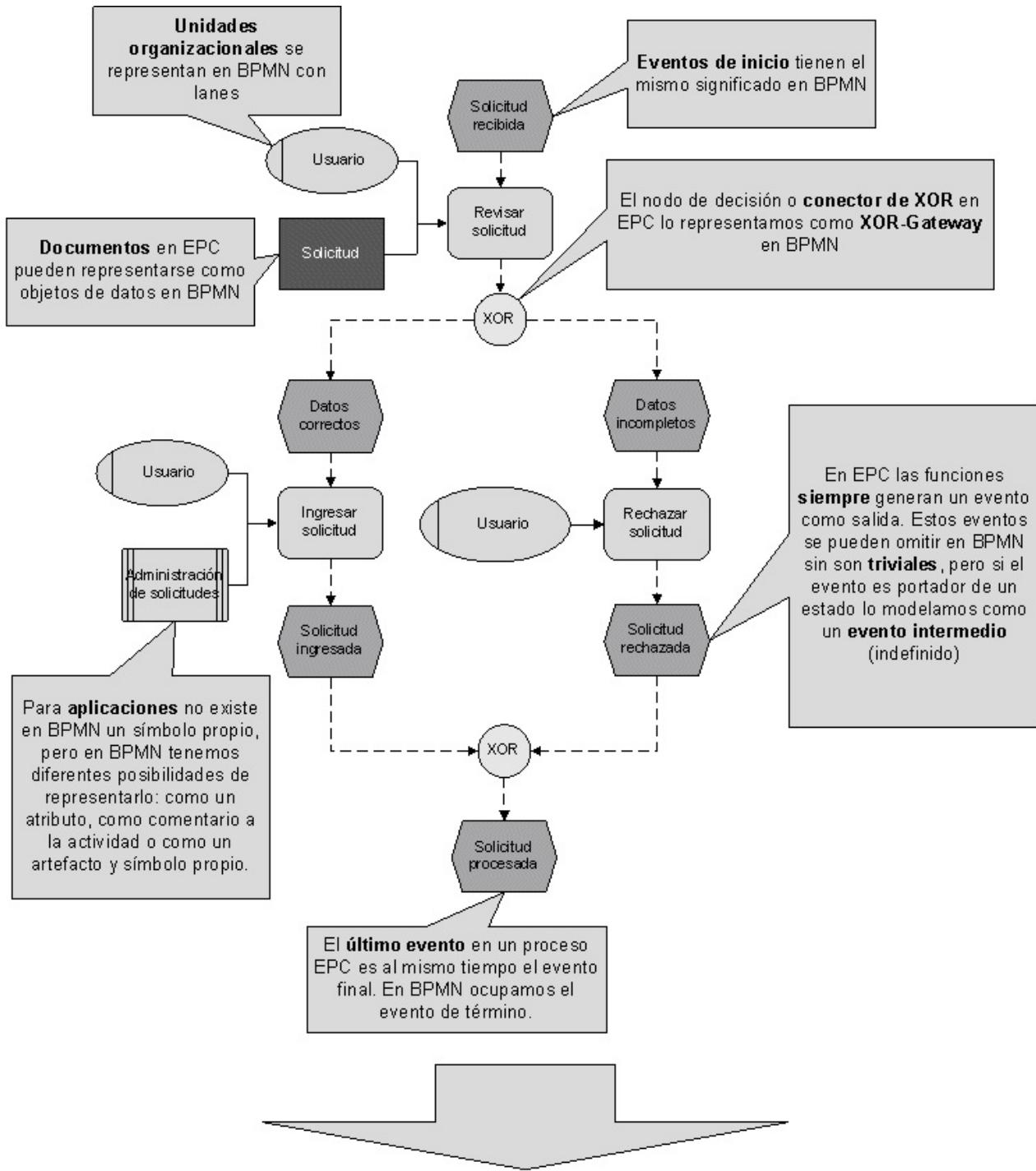


Figura 2.116: Mapeo de EPC hacia BPMN

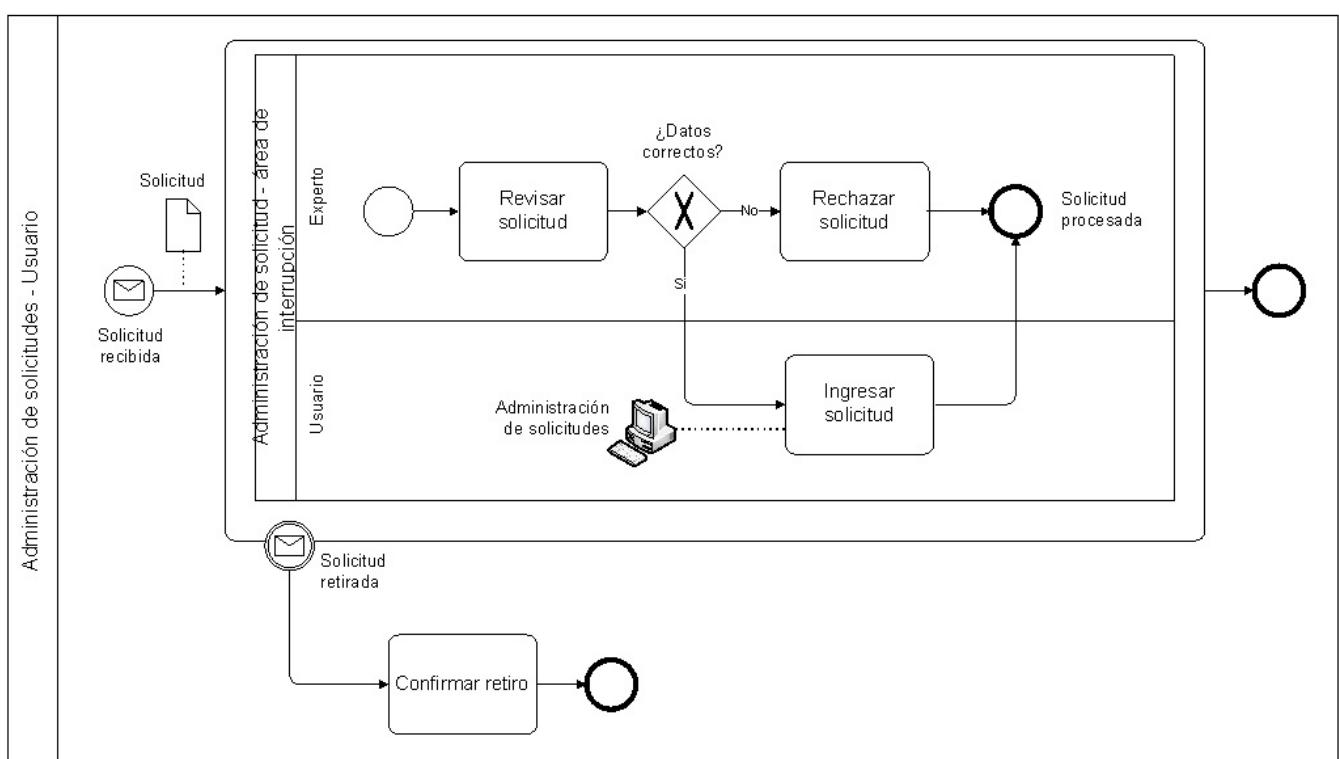
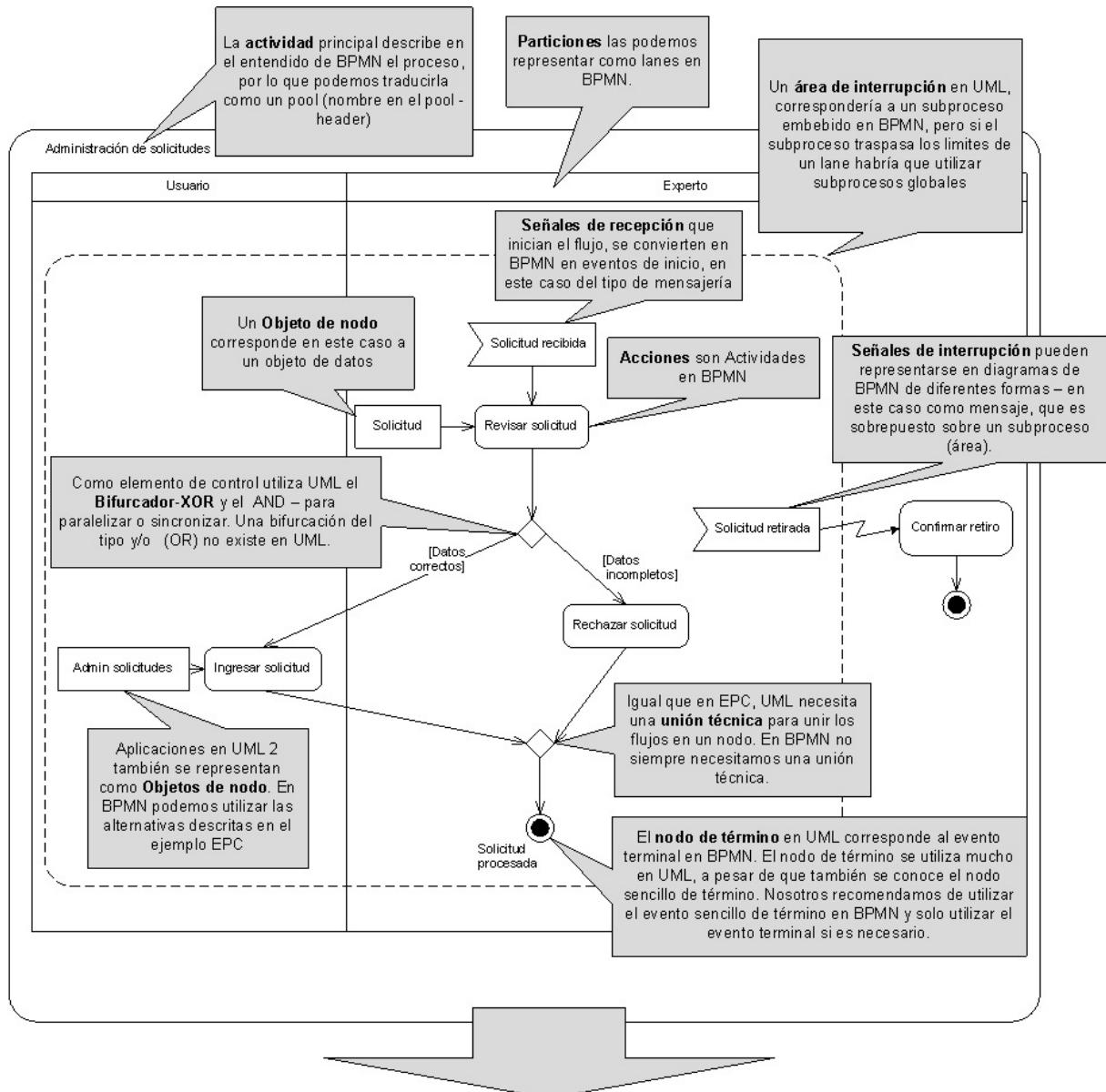


Figura 2.117: Mapeo de UML diagrama de actividades a BPMN

2.13.5. Indicadores y probabilidades

En muchas ocasiones analistas que han trabajado con herramientas como ARIS nos preguntan si pueden correr simulaciones para diferentes indicadores en BPMN como consumo de recursos, tiempos de ciclo, etc.. La respuesta a esta pregunta es que estas técnicas de análisis no son parte de una metodología de modelamiento de procesos. Las funcionalidades de simulación y formulación de indicadores son independientes de estas notaciones, pero pueden complementarse o hacer uso de los procesos documentados en un notación determinada. Para hacer uso de estas funcionalidades, el punto principal son las herramientas. Al igual que en EPC, IDEF-0 o UML Activity diagram se pueden hacer correr simulaciones con BPMN, analizando diferentes indicadores, siempre y cuando la herramienta lo soporte.

También tenemos que aclarar que ninguna especificación de BPMN hace alguna observación respecto a este tema o a los tipos de atributos que son necesarios para ampliar la notación con respecto a la documentación de funcionalidades complementarias. Por el momento el estándar no considera una ampliación del contexto en que opera, pero tampoco prohíbe que los proveedores en su oferta tecnológica lo hagan.

En la figura 2.118 hemos construido, a modo de ejemplo, un caso simple con un proceso diagramado en BPMN que incluye indicadores como tiempo de espera (TE), tiempo de procesamiento (TP), duración de ciclo (DC) y probabilidades. Si entra una solicitud, en promedio no será revisado antes de 2 horas. El ejecutivo se demora por lo general 15 minutos en revisar la solicitud y debe tomar como resultado de este procedimiento la decisión si va rechazar la solicitud o la va a aprobar para que la ingresen en el sistema de administración de solicitudes. En 90 % de los casos, es decir en 9 de cada 10 solicitudes, los datos están correctos y los formularios son aprobados. Bien, ahora queremos saber cuánto se demora en promedio (DC) hasta que una solicitud sea aprobada o rechazada. Este indicador no lo podemos retener en la notación si no incluimos un objeto que lo permita. El objeto que debemos ocupar para estos fines en BPMN, es el evento intermedio indefinido, al cual le agregamos el atributo que retiene el indicador. Para nuestro ejemplo tendríamos una duración de ciclo de 160 minutos para que la solicitud haya sido aceptada o de 140 minutos para los casos en que sea rechazada. Si consideramos las probabilidades, obtenemos una duración promedio de ciclo general de 158 minutos, para el proceso que lógicamente lo retenemos en el evento final.

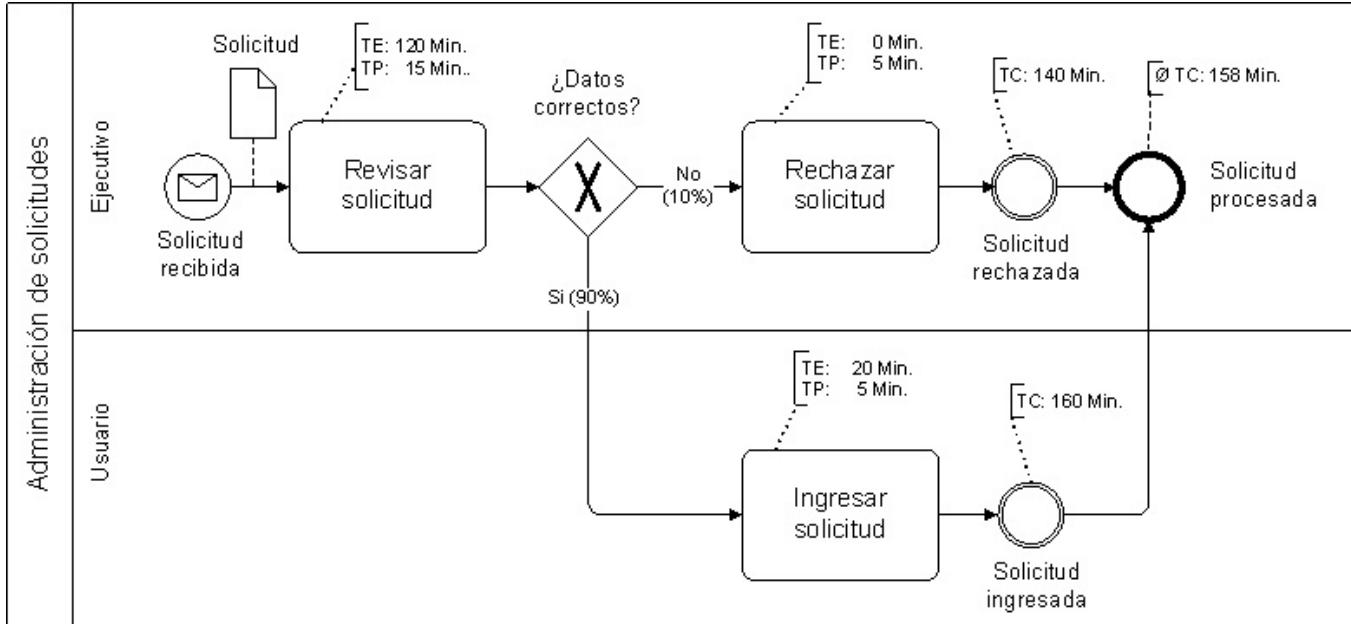


Figura 2.118: Ejemplo de representación de indicadores en BPMN

En el capítulo 6, secciones 6.22 y 6.23 describiremos como podemos ampliar el dominio de BPMN a técnicas de análisis y mejoras, relacionadas con la optimización de tiempos de ciclo y costeo de actividades. Con el objetivo de aclarar una discusión muy frecuente sobre qué plataformas abarcan más funcionalidad que sólo BPMN (como ARIS, Process-Modeler u otras), es necesario enfatizar que no tiene ninguna relación una notación para un fin determinado, como lo es BPMN, con aspectos complementarios o relacionados con ella. No confundamos que ARIS, Idungu u otras herramientas ofrecen muchas funcionalidades que complementan BPMN e implementan otra notación adicional como IDEF, UML o EPC. Al entorno para administrar estos modelos inter-relacionados se le llama «Arquitectura Empresarial»

Capítulo 3

Nivel 1: Modelo de Procesos Descriptivos

3.1. Sobre este Nivel

3.1.1. Objetivo y Beneficios

Un modelo de proceso en el primer nivel describe la lógica de negocio lo más compacta posible. El objetivo principal en este nivel es describir el alcance que tienen los procesos de principio a fin (ver figura 3.1).

El lector puede reconocer sin mayor complicación a quién está dirigido el proceso, cuales son las principales actividades y que tipo de productos y servicios son los creadores de valor. En algunas ocasiones los diagramas del primer nivel pueden enriquecerse de mayor información si se considera apropiado, como sistemas que apoyan los procesos.

El primer nivel sirve como introducción y definición de alcance a todos los participantes, pero principalmente a los responsables de áreas o procesos en general como el Process Manager o Process Owner.

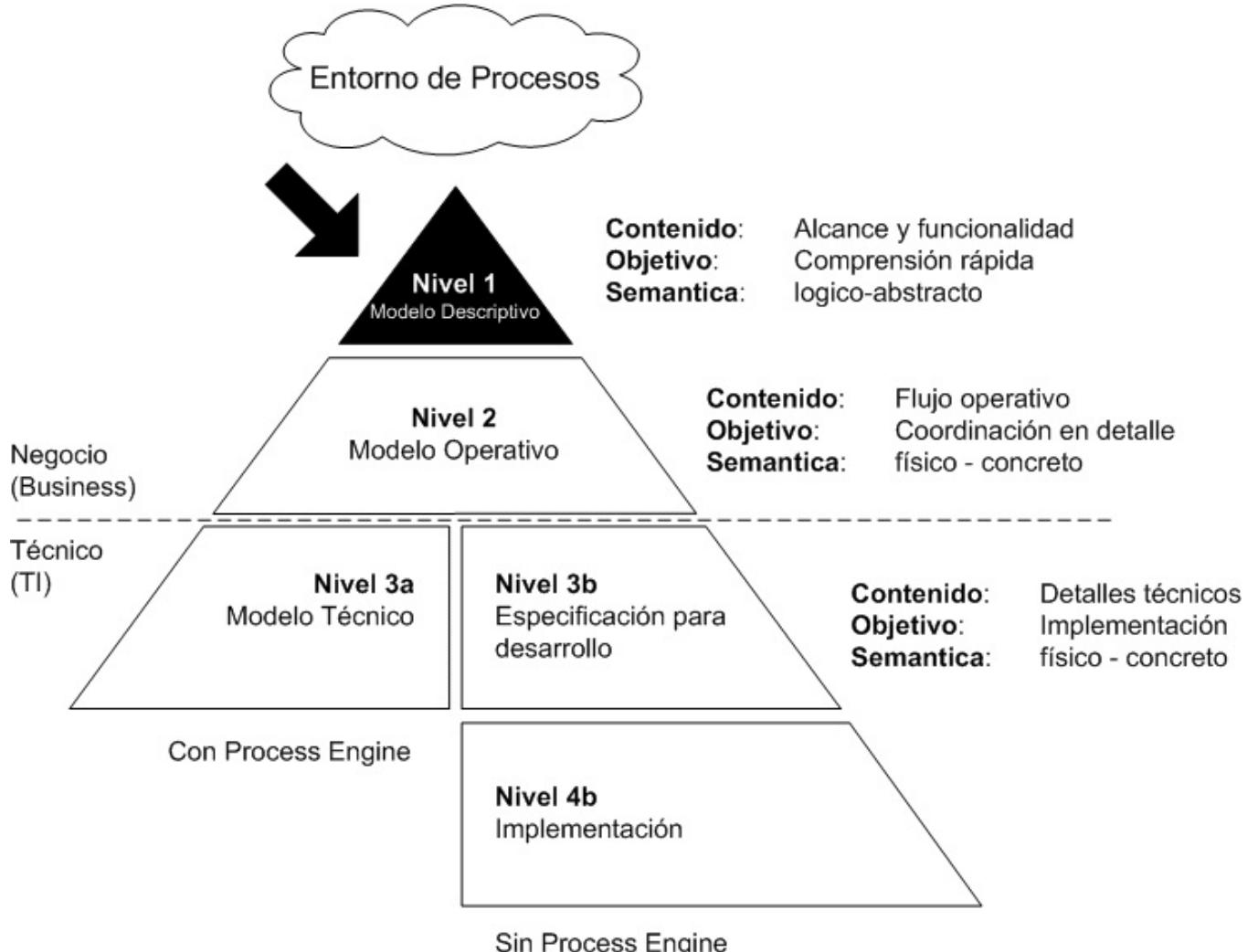


Figura 3.1: Nivel 1 Descriptivo del Marco Estructural para BPMN

Los principales objetivos relacionados con el primer nivel son:

Definición del alcance de los procesos (Límites definidos en los términos desde/hasta)

Asignación de las responsabilidades y recursos del proceso

Definición de los principales KPI's (Key Performance Indicators: Indicadores críticos del negocio), por ejemplo tiempos de ciclo máximo por proceso

Requerimientos generales que se esperan para mejorar el rendimiento de los procesos

3.1.2. Requerimientos para el modelo

Con el objetivo de poder cumplir con los requerimientos anteriormente mencionados en el primer nivel, el modelo debe entenderse fácilmente. El modelo presentado también debería poder interpretarse por personas que no tienen conocimientos de BPMN. Para el diseño de sitios web, existe un excelente libro escrito por Steve Krug, cuyo título o slogan también podríamos utilizarlo como principio para diseñar el nivel 1 de nuestro Framework de BPMN:

¡Don't make me think ! («no me hagas pensar»)

Este refrán puede que le parezca algo exagerado, pero no lo es. En el nivel descriptivo debe reconocerse claramente cual es el valor que se genera para el cliente. De acuerdo a los principios fundamentales de la gestión por proceso, existen los procesos sólo porque están destinados a crear valor para el cliente y muchos de estos potenciales o prestaciones se definen para satisfacer la necesidades de los clientes. En muchas ocasiones son justamente estos potenciales los que se analizan para mejorar los niveles de eficiencia en proyectos de BPM.

Ningún proceso puede captarse completamente en una vista resumida, pero si podemos esforzarnos en abstraer lo más importante y resumirlo en un diseño gráfico de no más de una hoja tamaño carta; ese es nuestro desafío en el nivel descriptivo. Al mismo tiempo esta forma compacta de representar un modelo se hace compatible con el formato de PowerPoint para presentaciones. Desde luego no debería intentar de representar el máximo de objetos y relaciones en un modelo tamaño carta, lo que lo convertiría igualmente en un proceso ilegible. Debido a lo anterior, recomendamos de no incluir más de 10 objetos de flujo y 8 artefactos en un modelo del nivel descriptivo.

Como en todas las cosas, siempre existen ventajas y desventajas si formulamos una norma. Si pretendemos desarrollar modelos de procesos fáciles de entender, no podemos utilizar toda la simbología que ofrece BPMN. No podemos pretender que se entienda en forma intuitiva lo que significa un evento de compensación o lo que significa una actividad del tipo instancia múltiple. El precio que tenemos que pagar al restringir la simbología es que perdemos gradualmente en expresividad, es decir el modelo es menos preciso. La especificación no reglamenta qué objetos o símbolos utilizar en modelos de mayores niveles de abstracción; eso queda a decisión suya. En la sección 3.3 le hacemos una proposición de los objetos, que a nuestro juicio se prestan para modelar el nivel descriptivo. También se puede dar la combinación de reducir por una parte los objetos estándar de BPMN y enriquecer la expresividad con artefactos propios del negocio como lo vamos tratar más adelante en esta sección.

El segundo compromiso que vamos a admitir en modelos del nivel descriptivo, es de desistir en completitud en cuanto a la representación de la lógica o de la semántica de los procesos. En la sección 3.2 vamos a mostrar a través de un ejemplo, que modelos de procesos en el nivel descriptivo en muchas ocasiones no son, o mejor dicho no pueden ser consistentes. La decisión de permitir esto, no siempre resulta fácil, pero por otra parte hemos tenido la experiencia que modelos descriptivos consistentes, no son entendidos por las personas a quienes están dirigidos (ejecutivos de alto nivel) porque son muy complejos. Entonces no logramos nuestro objetivo y estos modelos pierden su razón de ser, razón por la cual admitimos el compromiso de permitir inconsistencias en el nivel descriptivo, pero sólo en el nivel 1. Al pasar al nivel 2 (operativo), no es aceptable que los modelos contengan

inconsistencias.

Con respecto a la sintaxis somos más exigentes en cuanto a la representación de modelos de procesos en el nivel descriptivo. La reglas estructurales (sintaxis) no se deben infringir en el nivel 1. Sólo en casos muy excepcionales hemos tolerado a beneficio de un mejor grado de expresión, pequeñas desviaciones de las reglas sintácticas en el nivel descriptivo.

Nuestro Tip en BPMN

En el nivel descriptivo rige el siguiente principio: En lo posible mantener las reglas sintácticas y si es necesario tolerar inconsistencias semánticas.

3.1.3. Técnicas de levantamiento en el primer nivel

¿Cuándo requerimos o se hace necesario modelar los procesos en el nivel descriptivo? Si tenemos que levantar por primera vez la situación actual de un proceso, si tenemos que diseñar un proceso nuevo o rediseñar uno existente (ver figura 3.2).

Levantar un proceso por primera vez, es más difícil de los que algunos se imaginan. En ocasiones contamos con algo de documentación, como procedimientos escritos, pero por lo general vamos a entrevistar directamente a los usuarios de negocio (process participants) o bien los encargados del negocio (Process Manager). Usted puede entrevistarlos en forma individual o reunirlos a todos juntos y realizar un workshop.

La ventaja de un workshop es de involucrar desde un principio a todos los participantes y darles la oportunidad de manifestar sus perspectivas del proceso, lo que debiera de aumentar el grado de aceptación del proyecto de BPM, pero por otro lado la dinámica de grupo puede complicar el proceso de levantamiento. Cada participante tiene su propia interpretación del flujo del proceso, se mezcla la

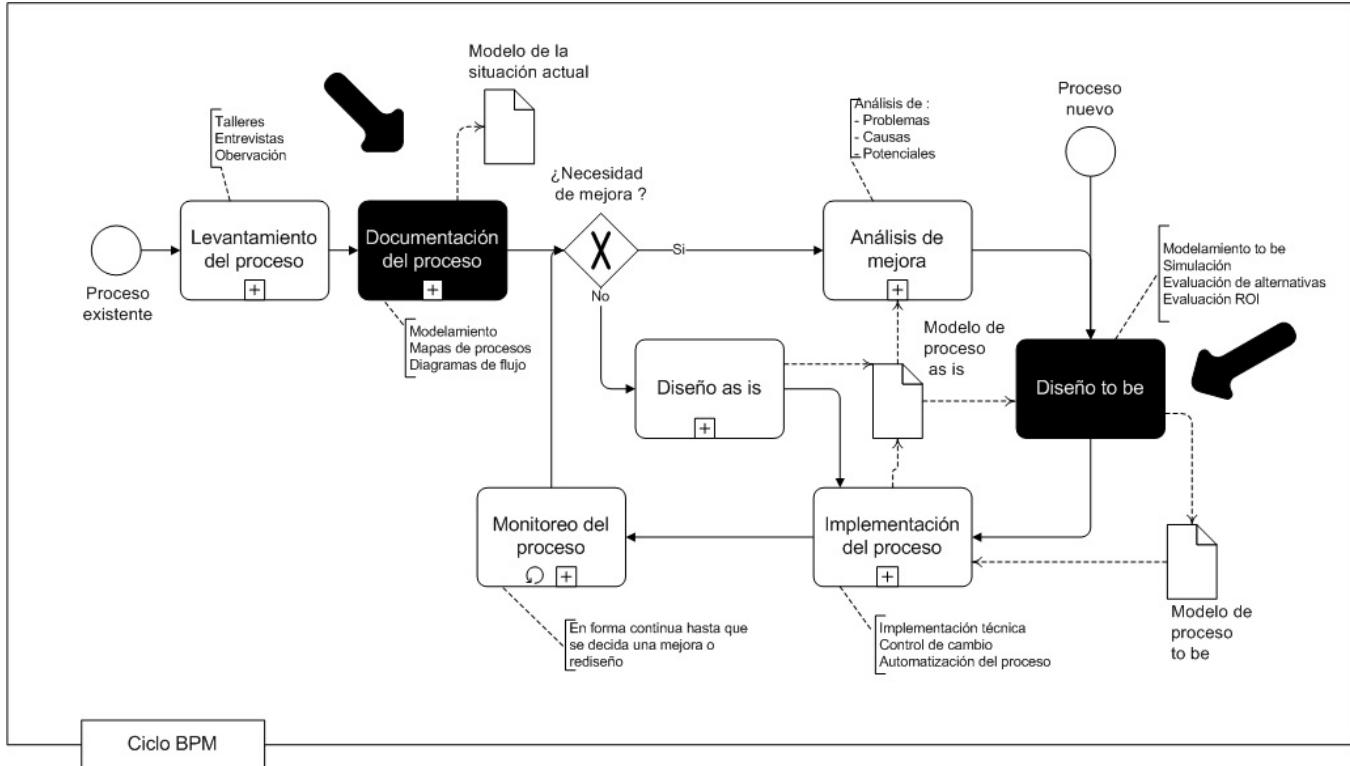


Figura 3.2: Modelos del primer nivel se levantan a partir de dos situaciones

forma en como cada uno ve el proceso. Algunos se van inmediatamente al detalle, otros quieren manifestar su descontento sobre las prácticas y si hay varias áreas representadas en el grupo puede tornarse rápidamente muy política la reunión. El moderador corre el peligro de perder rápidamente el control de la reunión y apenas haya dibujado dos cajitas en la pizarra, comienzan los comentarios:

«Antes de fijar la fecha de entrega, tenemos que revisar las condiciones de la orden de compra.»

«Pero eso no sucede siempre cuando recibimos la orden. A veces tenemos que evaluar la credibilidad del cliente.»

«Pero sólo si el volumen de venta es superior a los \$ 300.000.»

«Y además sólo si no es un cliente VIP.»

«Ciento, también habría que revisar ese caso, y ¿quién lo hace?»

«El ejecutivo de venta, quien otro.»

«En nuestro caso lo revisa el asistente, a nuestro ejecutivo por lo general no le queda tiempo para esa tarea.»

«¿En serio? ¿está permitido que el asistente haga esta evaluación? En nuestro caso, por lo menos siempre tiene que visarlo el ejecutivo.»

Podríamos seguir. Seguramente esta dinámica le parece conocida a cualquier analista que haya levantado procesos en reuniones con muchos participantes. Lo que queremos al principio es levantar sólo una vista general del proceso y abstraer de

los detalles, casos de excepción y malas prácticas. Los usuarios generalmente están sumergidos en el detalle del día a día y les cuesta abstraer lo importante del proceso. Si no actuamos con personalidad y buenas técnicas de moderación, corremos el peligro de terminar con un modelo mal logrado con una mezcla de detalle y vistas generalizadas, que puede que parezca completo, pero no lo es, y además contenga errores. Los participantes van a salir de la reunión estresados y mal humorados. Si le sucede esto, puede enterrar su proyecto de BPMN.

Siempre que usted inicie un workshop de levantamiento, debiera de asimilar la siguiente oración:

«¡Todos los modelos de procesos son incompletos - pero algunos tienen sentido!»

Este principio lo deducimos del matemático estadístico George E.P. Box y se refiere que nunca debiéramos intentar de captar o modelar todo el detalle desde un principio; sencillamente no resulta.

Al iniciar un workshop de levantamiento, usted tiene que comunicar claramente que el objetivo de la primera reunión es de levantar una vista muy generalizada y resumida del proceso. Para esta primera iteración usted declara los siguientes objetivos:

Queremos definir el proceso de principio a fin.

Queremos captar el proceso en no más de ocho pasos.

Sólo queremos representar el flujo normal.

Sólo queremos identificar los roles regulares que intervienen en el proceso. No queremos levantar aún ineficiencias, problemas o propuestas de mejora.

Si usted aclara al principio de la reunión estas condicionantes, estará en condiciones de levantar un proceso de negocio en forma resumida en no más de 30 a 45 minutos, pero tiene que ser perseverante cuando alguno de los participantes se desvíe y caiga 3.2. CASO DE ESTUDIO: PROCESO DE CONTRATACIÓN DE PERSONAL137

en el detalle. Cuelgue en forma visible y permanente estos objetivos en una pared de la sala de reuniones.

La primera reunión (kickoff) tiene también una importancia sicológica. Los integrantes van a salir de la reunión con una sensación de haber logrado algo y que se puede observar su proceso de negocio desde una perspectiva aérea. Entonces usted se queda con una base, un proceso en el que se ha definido su alcance de principio a fin pasando por las principales etapas. Con esta base usted puede repartir el trabajo y meterse en profundidad sin que se desestabilice el fundamento.

¿Se recomienda usar en esta primera sesión la notación de BPMN? En un principio si es posible, puede también servir para explicarle a los participantes los principales

objetos de flujo, pero no es tan importante, también se puede usar otro tipo de técnicas de moderación, como por ejemplo tarjetas adhesivas en la pizarra.

3.2. Caso de Estudio: Proceso de Contratación de Personal

Roberto es gerente del área de Recursos Humanos (RRHH) de una empresa. Él se ha propuesto mejorar y en general agilizar el proceso de contratación de personal de su empresa. Él piensa que sus usuarios administran muchas tareas en forma manual y que la eficiencia de estas actividades pueden mejorar significativamente con el apoyo de un sistema de workflow. Él tiene en mente las permanentes quejas de sus clientes (las áreas) de lo lento y burocrático que es actualmente el proceso de contratación, desde el aviso de un puesto vacante hasta la contratación de un profesional. Roberto está convencido que gran parte del tiempo se pierde porque los jefes de área se toman mucho tiempo en la evaluación de los candidatos preseleccionados y ante preguntas por parte del equipo de RRHH existe poca disponibilidad por parte de las áreas de responder en el tiempo adecuado, una suposición que no puede comprobar con datos duros.

Estamos reunidos con Roberto en la sala de conferencia conversando sobre su situación. Él describe el proceso de contratación de personal de la siguiente forma: «Si un área quiere ocupar un nuevo cargo o cargo vacante, el jefe me envía un e-mail con el aviso y la descripción de cargo que necesita junto a un formulario excel y los requerimientos de perfiles, competencias y...»

En este momento interrumpimos a Roberto y le hacemos a entender que no es objetivo de esta reunión de revisar los campos que tiene que llenar el jefe de área en el formulario de excel, sino que nos interesa primero levantar el flujo normal y resumido. Todo el detalle del proceso lo veremos más adelante.

«Ah, OK, está bien, entonces el jefe de área me avisa el cargo vacante por email, entonces yo tengo que revisar quién de mis usuarios tiene disponibilidad para atender el caso. Eso depende de la carga de trabajo que tenga cada uno, el cual reviso en el plan semanal de tareas de mis usuarios. Además me acerco a ellos a preguntarles como han avanzado en sus tareas pendientes.»

Nuevamente tenemos que frenar a Roberto en sus intentos de explicarnos como trabaja para asignar sus tareas a sus empleados. «Roberto: sólo nos interesa captar el flujo normal del proceso y filtrar todos los detalles que suceden en él» Él se muestra algo consternado, pero sigue:

«Bueno, entonces el procedimiento es muy sencillo: Licitamos el cargo y esperamos las postulaciones de los candidatos. Estas las revisamos, evaluamos, seleccionamos un candidato y ocupamos el cargo. En un principio nuestra misión termina, cuando el candidato seleccionado firma el contrato de trabajo, aún cuando nos queda por ingresar todos sus datos en nuestro sistema de RRHH, pero seguramente esto sería ya mucho detalle para ustedes ¿no cierto?»

Así es, a nosotros nos interesa levantar la siguiente información clave con respecto

al proceso:

El motivo por el cual un área tiene el requerimiento de ocupar un puesto o un cargo. El cargo es licitado, los candidatos postulan al cargo, las postulaciones son evaluadas, el cargo es ocupado.

El proceso llega a su término, cuando se firma un contrato de trabajo y se ocupa el cargo.

Con esta información modelamos el proceso que usted aprecia en la figura 3.3. Roberto valida este modelo sin ningún problema. Sólo el evento del tipo de condición, fue un objeto técnico que nosotros introdujimos al diagrama y que le explicamos en pocas palabras a Roberto. El evento de término lo pusimos intencionalmente en el lane del área, con la finalidad de respetar el principio en BPM, que cada proceso es impulsado por el cliente y termina en el cliente.

Como experto en BPMN le tiene que haber resaltado una inconsistencia semántica en este modelo. Imaginémonos que vamos a aplicar el concepto de token a este proceso, entonces tendríamos un problema con las actividades «enviar postulación» y «evaluar postulaciones». Si se envía sólo una postulación (singular), no podemos evaluar varias postulaciones (plural). Diríamos que esto es una contradicción, lo que nosotros denominamos técnicamente como inconsistencias semánticas.

El problema no se soluciona al cambiarle el nombre a la actividad por «enviar postulaciones», porque en este caso se interpretaría como que el postulante envía varias postulaciones a una misma licitación de cargo. ¿Qué hacer? Si queremos

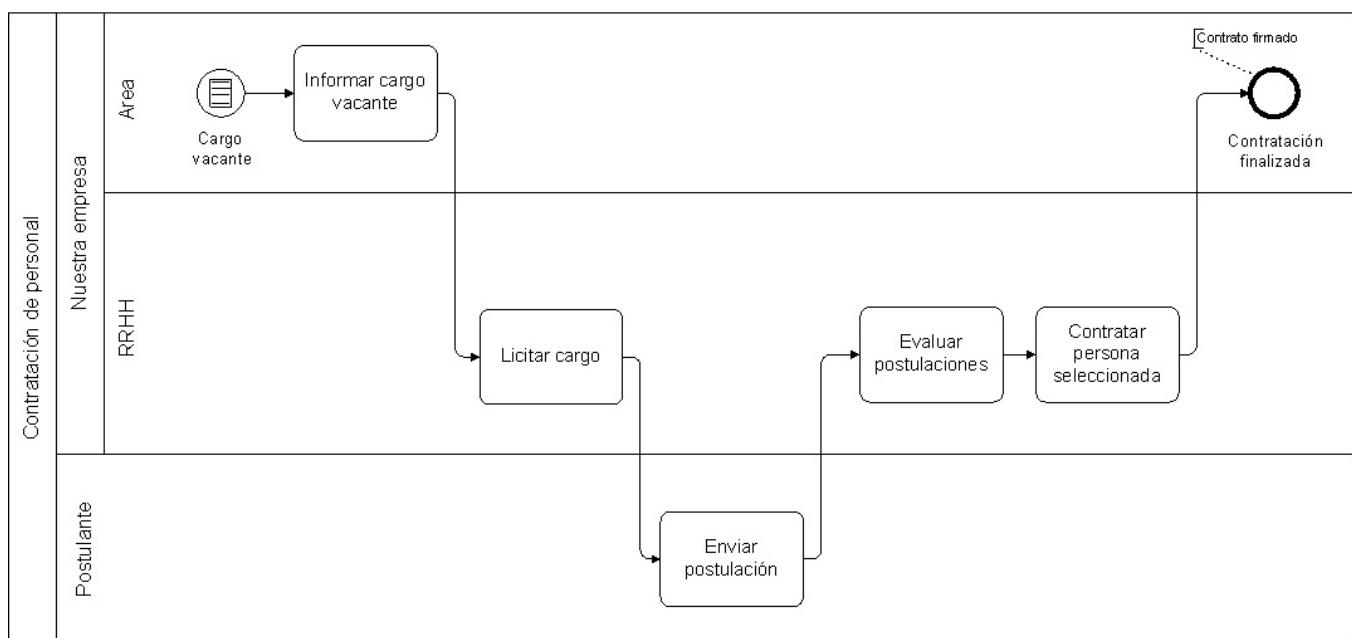


Figura 3.3: Modelo de proceso de contratación de personal para nivel descriptivo
mantener el modelo en forma generalizado y resumido, no conocemos una solución

para este problema.

¿Qué diría Roberto a este problema? Seguramente nada, porque el no detecta ningún problema, el modelo es entendible y sabe como están relacionadas las actividades. Si el modelo abstraído nos sirve para delimitar el contexto y definir el alcance, hemos logrado nuestro objetivo en el nivel descriptivo y aceptamos desde el punto de vista formal la inconsistencia semántica.

Queremos hacer incapié que el modelo tiene otra inconsistencia semántica: No está especificado que las áreas también deberían participar en la evaluación de los candidatos. Justamente este es uno de los puntos que Roberto, gerente de RRHH, reclama sobre la poca disponibilidad de los jefes de áreas. También tomamos conciencia sobre esta inconsistencia, porque aún no hemos entrado en proceso de análisis. Si en el nivel 1 nos encontramos con el caso que una actividad es compartida por varios participantes, le asignamos la tarea a aquél participante (lane) que es responsable final de la ejecución de esta.

3.3. Restricción de la Tabla de Tipos de Objeto (Simbología)

La norma de BPMN cuenta con más de 50 símbolos, los que usted conoció en su totalidad en el capítulo 2 de este libro. En el nivel descriptivo no necesitamos tantos símbolos y no podemos esperar que todos los usuarios conozcan la taxonomía del estándar BPMN, razón por la cual nosotros utilizamos sólo un subconjunto de todos estos elementos. La especificación no dice nada al respecto, pero nosotros le vamos a hacer una propuesta. Lógicamente que usted puede definir para su organización su propio subconjunto, pero de todas formas le recomendamos que haga una convención de filtro y no permita a los analistas usar al azar cualquier elemento en el nivel descriptivo.

3.3.1. Pool y lanes en el nivel descriptivo

Si usted leyó con detención la sección 2.9, debería haberse dado cuenta que la representación del modelo que presentamos en nivel 1 (ver figura 3.3) tampoco es correcta en cuanto a la forma como se modela con pool y lanes en BPMN. La especificación dice que cada pool tiene su propio dirigente y nosotros representamos a todos los participantes en un pool subdividido en lanes. Este dirigente no existe en este proceso de la figura 3.3, pues tampoco es orquestado por un motor de workflow. Normalmente deberíamos de modelar el aviso como un flujo de mensaje que le envía el jefe de área a RRHH y separar el pool del área del de RRHH, como lo muestra la figura 3.4. De acuerdo a este modelo, cuando se produce una vacante el área envía vía flujo de mensaje el aviso a RRHH y cuando el cargo ha sido ocupado, se le informa al área que el candidato ha firmado el contrato de trabajo.

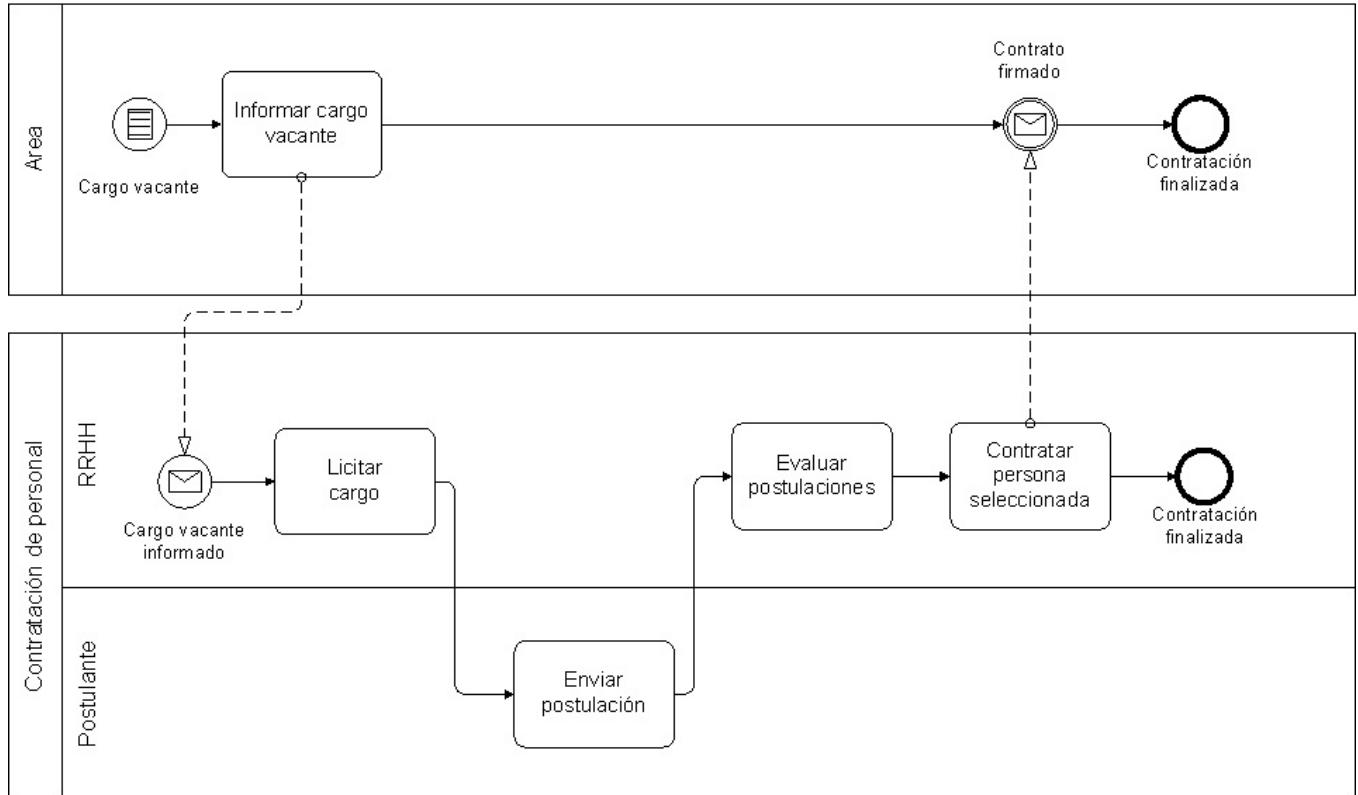


Figura 3.4: Modelo de proceso de contratación de personal con Lane de área convertido en Pool

A primera vista esta representación tiene su gracia, pero sigue siendo un problema: Los postulantes también deberían representarse en un pool propio, porque son externos a la organización y tienen su propio dirigente. En la figura 3.5 mostramos este modelo como un diagrama de colaboración, en el cual cada participante recibe su propio pool.

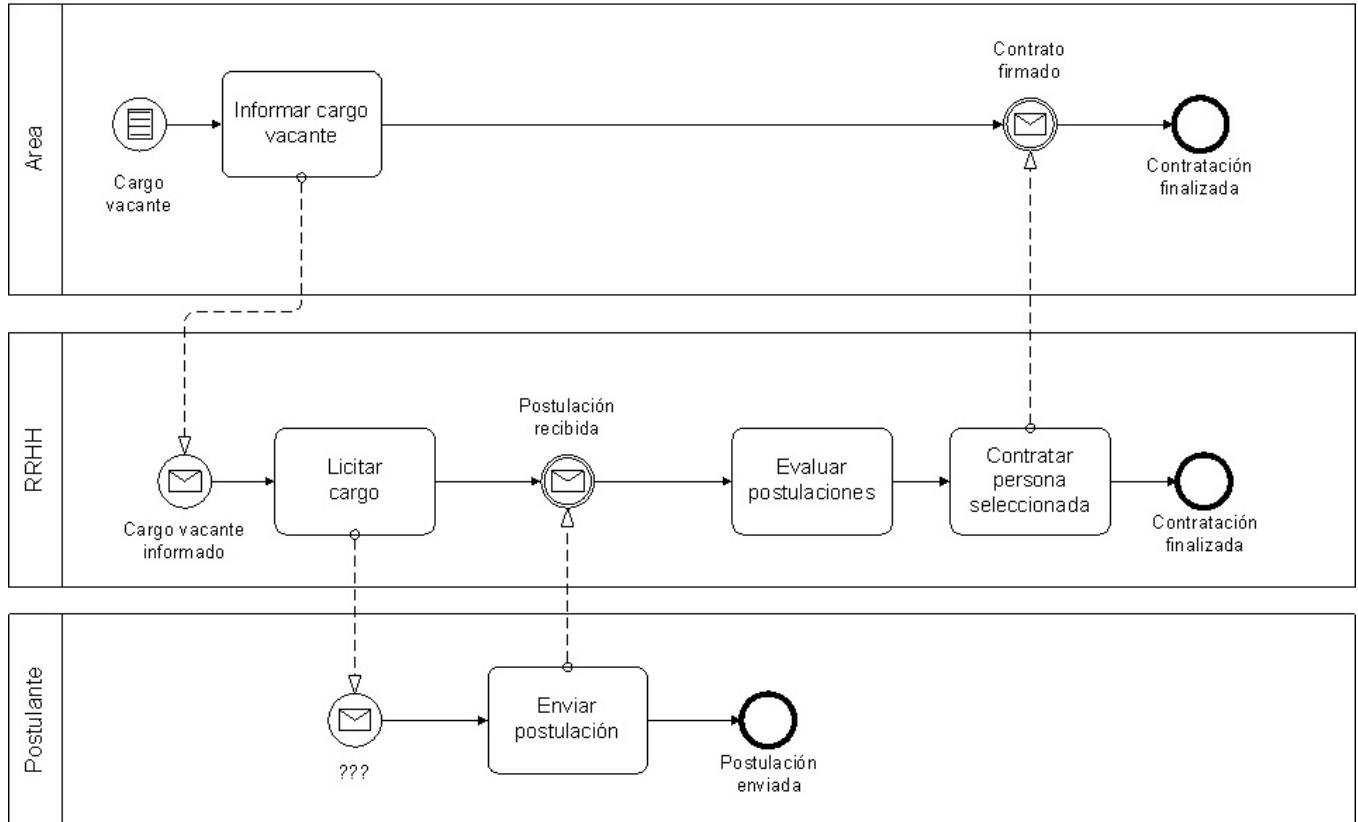


Figura 3.5: Modelo de proceso de contratación de personal con cada participante en Pool propio

El signo de interrogación en el pool del postulante nos revela que mientras más lógica representamos en el nivel descriptivo, más preguntas nos van invadiendo y por consiguiente mayores inconsistencias va adquiriendo el modelo. Por ejemplo ¿cómo informamos al candidato que fue seleccionado? En la realidad existe aquí todo un subproceso, de postulación, invitación a entrevistas etc., y el canal por el cual postula. Si el postulante ve la licitación en una bolsa de trabajo deberíamos iniciar el proceso del postulante con un evento del tipo señal y no del tipo de mensaje como muestra la figura 3.5. Por otro lado tenemos el problema que el flujo no espera sólo una postulación, sino varias, lo que tampoco muestra el modelo. El modelo tampoco nos revela si vamos a esperar un tiempo a que entren todas las postulaciones o las vamos a ir evaluando a medida que vayan llegando.

Como usted puede apreciar con este ejercicio, no sabemos donde detenernos al preguntar cuanta lógica representamos en el nivel descriptivo. Habiendo avanzado con este análisis podríamos modelar todo el detalle con el gerente de RRHH, pero ¿es el objetivo del nivel 1? ¡Claramente no! Cuando le mostramos el diagrama de la figura 3.4 a Roberto, arrugó el seño. Con algunas explicaciones nuestras logró entender el modelo, pero ¿sería igual si le mostramos el modelo de la figura 3.5? Ahora va a detectar inconsistencias, porque desde su punto de vista va a estar incompleto y si expandimos el modelo y agregamos mayor simbología, no lo va a entender, no se va a querer dar el tiempo para entenderlo y por ende no lo va a aceptar.

Esta es la razón por la cual no modelamos en el nivel 1 lógica como lo muestra la figura 3.5 y lo guardamos para seguir descomponiéndolo en el nivel 2.

Por las razones anteriormente expuestas, desistimos «por lo general» en el nivel 1 de usar diferentes pools. En algunas ocasiones hacemos una excepción, por ejemplo un cliente externo que interactúa con el proceso de la empresa. En este caso escogimos un pool cerrado para marcar el hecho, que el cliente es externo y mostrar la lógica del proceso en un propio pool desde el punto de vista de la empresa, que es lo que se quiere mostrar. Este caso se utiliza mucho como ejemplo didáctico para explicar el concepto de colaboración. La figura 3.6 muestra este ejemplo que se entiende sin mayor explicación. Sería fantástico si pudiéramos reducir todo tipo de proceso de negocio en el nivel 1, como lo muestra la figura 3.6, pero la realidad nos muestra que esto no es siempre así.

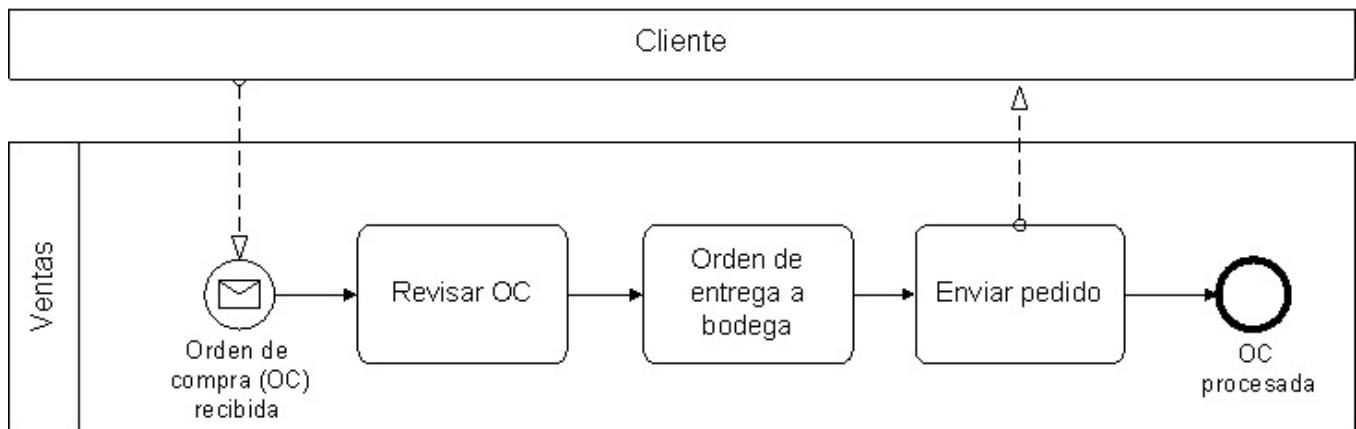


Figura 3.6: Caso de modelo de colaboración utilizando Pool cerrado

En el caso de nuestro proceso de contratación de personal, tenemos en la realidad muchos más actores como agencias de empleo, postulante y además clientes internos (áreas). ¿Bajo qué criterio escogemos el pool más importante para separarlo?

3.3.2. Actividades y subprocessos en el nivel descriptivo

La siguiente pregunta que nos debemos hacer con respecto al nivel descriptivo es sobre el uso de actividades y subprocessos. Podríamos reducir el primer nivel sólo a la representación de subprocessos, pero la realidad nos muestra que no siempre lo podemos hacer. En el nivel 1 desistimos de diferenciar en tipo de actividades (ver sección 2.7.1), o de utilizar los marcadores (ver sección 2.7.2), sólo con una excepción, la del tipo «loop». Esta marca es relativamente intuitiva y se entiende sin mayor explicación.

Nuestro Tip en BPMN

En la sección 2.2 hicimos una propuesta de cómo denominar actividades: «La acción sobre un objeto», es decir el nombre de una actividad siempre está compuesta por un verbo y un objeto, por ejemplo «enviar postulación». Para denominar subprocessos en nivel 1 sustantivamos ambos elementos, por ejemplo la actividad «licitar cargo» la denominamos «licitación de cargo» y «evaluar postulaciones» en «evaluación de

postulaciones» al convertirla en un subprocesso. En algunos casos resulta algo ficticio la converción como el ejemplo de «enviar postulación» a «envío de postulación». En estos casos podemos desistir del verbo porque «postulación» se entiende como un subprocesso. Al mantener esta convención logramos la ventaja de diferenciar una actividad de un subprocesso, que encierra toda una lógica propia. De esta forma podrá evitar discusiones frecuentes que se dan, de por qué una actividad la denominamos una forma y un subprocesso de otra, al no contar con convenciones como las que les proponemos nosotros.

Subprocesos representan una lógica abstraída en procesos complejos y se prestan muy bien para representar un conjunto de actividades que representan una finalidad delimitada. En nuestro modelo de contratación de personal podemos declarar las actividades «licitar cargo», «enviar postulación» y «evaluar postulaciones» como subprocessos, porque intuimos o sabemos que son complejas . Sin embargo la primera actividad «informar cargo vacante», parece reducirse a llenar un formulario de excel y enviarlo a RRHH, es decir una actividad simple que no se sigue descomponiendo (atómica), razón por la cual no la transformamos a un subprocesso.

Si transformamos nuestro modelo agregado del nivel 1, considerando la diferenciación entre actividades y subprocessos, obtenemos un modelo como lo aprecia en la figura 3.7.

La siguiente pregunta es si es necesario modelar la lógica de los subprocessos que se encuentran cerrados. Por lo general desistimos de ello en el nivel descriptivo, porque no es objetivo del nivel 1 conocer los detalles de la lógica de negocio. Se suma además el hecho, que no podemos encontrar un nivel intermedio, sin llenarnos de inconsistencias semánticas como tratamos en la sección anterior.

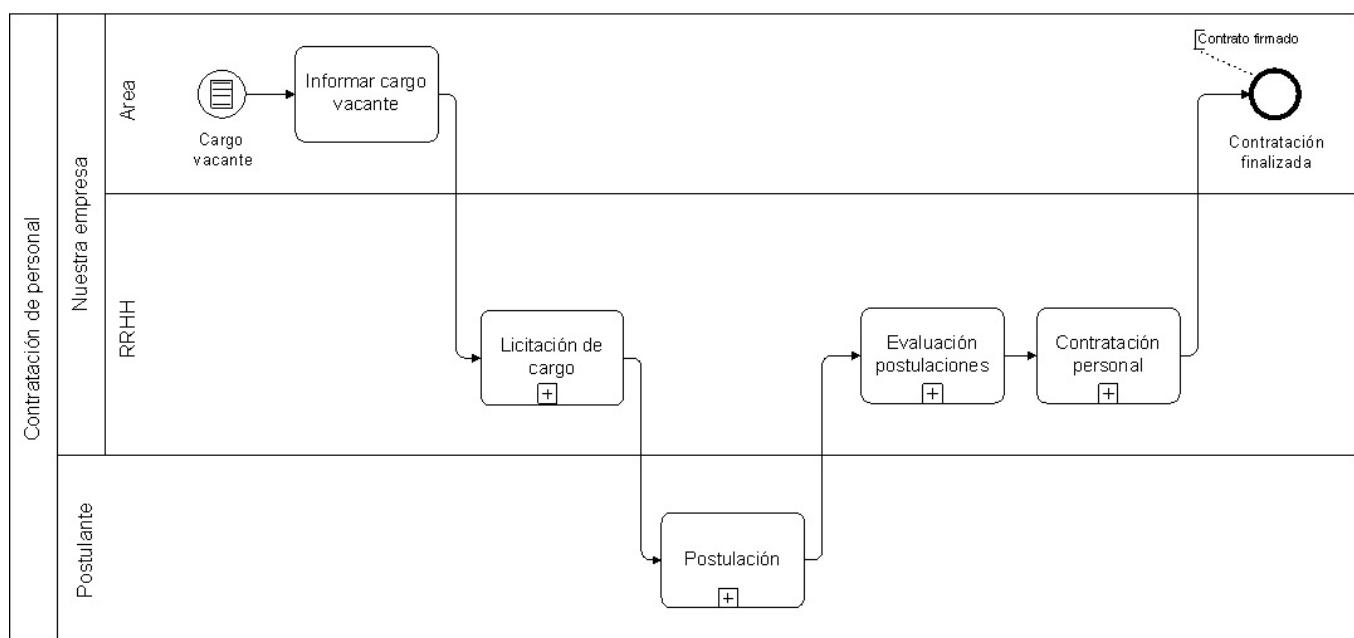


Figura 3.7: Modelo de proceso de contratación de personal con actividades y

subprocesos en nivel descriptivo

3.3.3. Gateways en el nivel descriptivo

Nuestro ejemplo de contratación de personal como lo muestra la figura 3.7, parte de la base que siempre encontraremos el candidato que queremos contratar, pero eso no es real. En ocasiones puede que no se presente o no encontremos al candidato con el perfil que buscamos. Ahora podríamos modelar que hacemos en esos casos y lógicamente lo haríamos utilizando Gateways, pero nosotros desistimos de esta casuística en el nivel 1. Lo que queremos representar en este nivel es el «happy path» o dicho en español el caso deseado o el caso normal. En la mayoría de los casos no es necesario mostrar en el primer nivel, las variaciones que puedan existir.

Dependiendo del proceso de negocio, puede que se haga necesario representar en este nivel flujos alternativos, por ejemplo si estos son parte del flujo normal. Este caso puede darse por ejemplo cuando un cliente puede escoger varios productos o si el inicio del proceso se da por diferentes eventos de entrada.

En estos casos recomendamos utilizar en el nivel 1 sólo los siguientes objetos de Gateways (véase también figura 3.8):

XOR-Gateways para los casos de bifurcación, pero evitamos de utilizar flujos condicionales que salen directamente de las actividades. Nosotros constatamos en la práctica que los XOR-Gateways son más intuitivos y se entienden mejor que los flujos condicionales.

No utilizamos para la unión técnica el elemento XOR-Join, es decir dejamos que los flujos entren directamente a la actividad siguiente, porque el lector no versado en BPMN podría preguntarse como funciona y que significa el XOR-Join, sobre todo en construcciones del tipo loop. Además obtenemos un beneficio adicional, el diagrama pasa a ser más compacto. En algunos casos puede darse la necesidad de utilizar la unión técnica, por ejemplo ante un evento intermedio, pero por lo general tratamos de evitarlo.

Con el AND-Gateway no siempre podemos ahorrarnos un elemento como en el caso anterior, porque ante el paralelizar un flujo, por lo general tenemos que sincronizarlo posteriormente, razón por la cual recomendamos usar el par (paralelizar y sincronizar).

No recomendamos utilizar el OR-Gateway, debido a que es un objeto complejo difícil de emplear y de entender. En un principio podemos representar siempre la lógica del OR-Gateway con una combinación de XOR y AND-Gateways, a pesar que el diagrama se expande más que con un sólo OR-Gateway, es más fácil de entender la lógica. De todas formas, tratamos de evitar de modelar una lógica del

tipo «OR» en el nivel 1.

Por ningún motivo utilizar el Gateway Complejo. Como el nombre lo indica se trata de un objeto de flujo que ni siquiera en forma gráfica es interpretable sin una explicación de texto, por lo que tampoco puede tener alguna justificación para utilizarlo en el nivel descriptivo.

	Recomendado	No Recomendado
Bifurcación	<p>Diagram of a recommended branching structure: a rounded rectangle leads to a decision diamond with an X. Two outgoing arrows are labeled "Respuesta 1" and "Respuesta 2".</p>	<p>Diagram of an unrecommended branching structure: a rounded rectangle has two outgoing arrows labeled "Respuesta 1" and "Respuesta 2".</p>
Unión	<p>Diagram of a recommended joining structure: two parallel incoming arrows merge into a single rounded rectangle.</p>	<p>Diagram of an unrecommended joining structure: a rounded rectangle leads to a decision diamond with an X, which then splits into two parallel outgoing arrows.</p>
Paralelizar	<p>Diagram of a recommended parallel structure: a rounded rectangle leads to a decision diamond with a plus sign. A feedback loop returns from the right side to the left side before merging back into the main flow.</p>	<p>Diagram of an unrecommended parallel structure: a rounded rectangle has two outgoing arrows.</p>
Sincronizar	<p>Diagram of a recommended synchronization structure: a decision diamond with a plus sign has a feedback loop returning to its input side.</p>	<p style="text-align: center;">Sin AND-Join Imposible de representar</p>
y/o	<p style="text-align: center;">Condición 1? Condición 2?</p> <p>Diagram of a recommended "y/o" structure: a decision diamond with a plus sign branches into two parallel paths. The top path leads to a decision diamond with an X labeled "Si". The bottom path leads to a decision diamond with an X labeled "ja". The "ja" path has two outgoing arrows labeled "nein" and another arrow pointing to the right.</p>	<p>Diagram of an unrecommended "y/o" structure: a decision diamond with a circle has two outgoing arrows labeled "Condición 1" and "Condición 2".</p>
Complejo	<p style="text-align: center;">Modelar diferente o no modelar en el nivel 1</p>	<p>Diagram of a complex structure: a decision diamond with a asterisk (*) has two outgoing arrows.</p>

Figura 3.8: Recomendaciones de uso de Gateways en el nivel descriptivo

3.3.4. Eventos y event Gateways en el nivel descriptivo

Con el objetivo de marcar el principio y el fin de un proceso recomendamos utilizar en el nivel descriptivo eventos de inicio y de término. Usted puede desistir del uso de eventos en el 1er nivel, entonces el modelo se vería como lo muestra la figura 3.9. La ventaja de desistir del uso de eventos es que se logra un diagrama más compacto, pero al mismo tiempo perdemos semántica. Ya no vemos que suceso inicia el proceso y que tiene que ocurrir para que termine en forma normal. Justamente una de las características que queremos representar en el 1er nivel, es el principio «end-to-end» y sin eventos es difícil de mostrarlo.

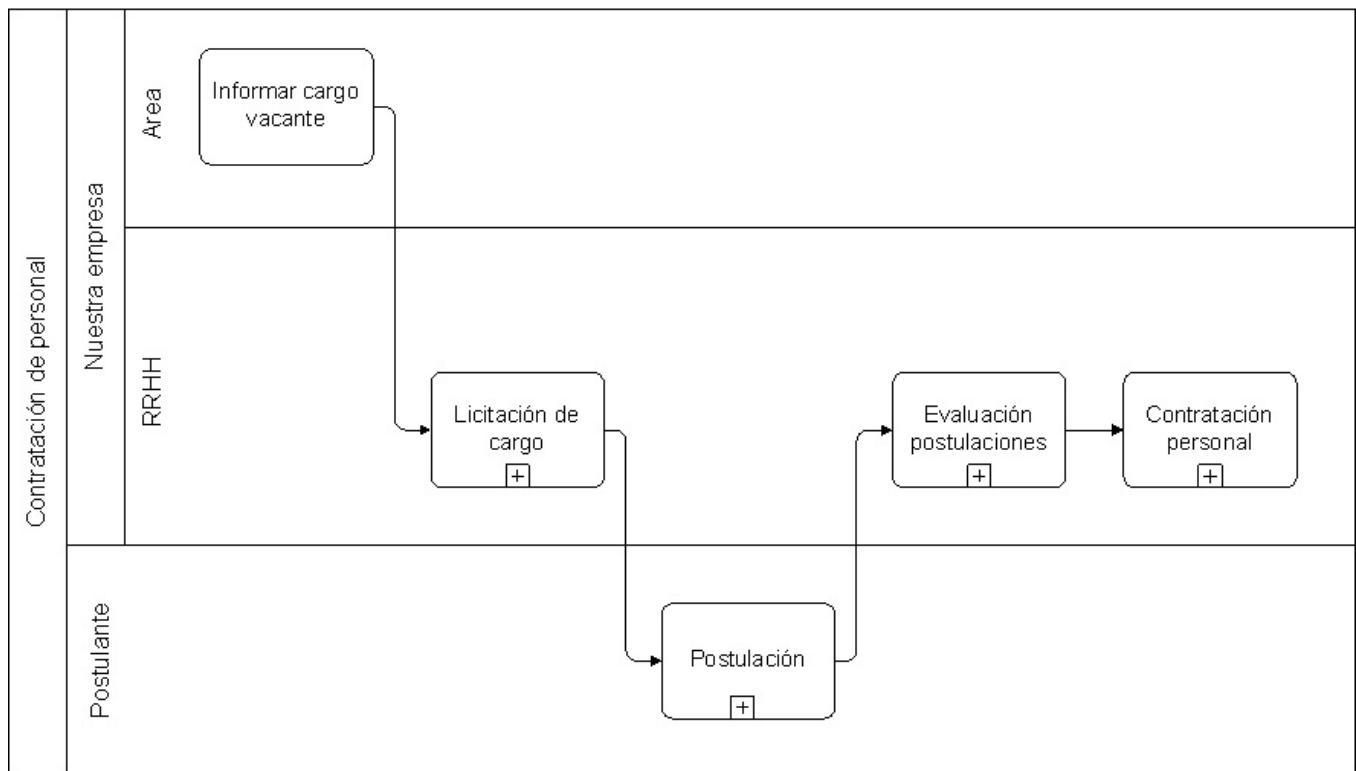


Figura 3.9: Modelo de proceso de contratación de personal sin eventos de inicio y de término en nivel descriptivo

En cambio los eventos intermedios son más complejos y por lo general requieren de una explicación para entender la funcionalidad que cumplen. A mucha gente le cuesta entender a primera vista que significado tienen los eventos intermedios, por ejemplo que el proceso tiene que esperar un suceso para que pueda continuar, razón por la cual ha de tratarse de explicitar lo mejor posible lo que significa un evento intermedio. Eventos intermedios de captura son más fáciles de entender que eventos intermedio impulsados, razón por la cual evitamos de utilizar estos últimos en el primer nivel, a excepción de los eventos indefinidos.

De acuerdo a nuestra convención sólo permitimos el uso de algunos eventos en el

nivel descriptivo:

Permitimos el uso de eventos indefinidos como eventos de inicio, intermedio y de término. El evento intermedio se presta para marcar un hito durante la ejecución del proceso. Estos estados intermedios pueden ser importantes para los responsables de los procesos que quieren marcar un hito, que se necesita en la gestión del proceso (monitoreo o control de gestión). En la figura 3.10 introducimos a modo de ejemplo dos hitos que marcan claramente dos etapas en el proceso, la licitación de cargo y la selección del candidato deseado, pero si tuvieramos que marcar un hito al cabo de cada actividad o subprocesso, ya no tendría sentido mostrarlo. En ese caso preferimos quedarnos con un diagrama más compacto y no modelar casos triviales.

Permitimos el uso de eventos de mensaje y de tiempo en el nivel descriptivo, tanto como eventos de inicio como intermedios, porque su simbología es bastante auto explicativa.

El evento condicional es un tanto más problemático, debido a que no se interpreta de inmediato, pero no lo queremos descartar en el primer nivel, porque algunos process manager, quieren ver que tipo de condiciones pueden gatillar o determinar la continuación de un proceso.

Por lo general nos esforzamos de mostrar el suceso que inicia un proceso de negocio, que tratamos de caracterizarlo como evento de mensaje, de tiempo o de condición. En la mayoría de los casos resulta, si no analizamos si realmente definimos bien el contexto, si es el caso ocupamos el evento del tipo indefinido. Puede ser que el dominio que estamos contemplando no abarca todo el proceso de negocio, razón por la cual no encontramos el evento de negocio que inicia el proceso. Para estos casos, el evento indefinido está bien escogido, puesto que nuestro proceso vendría a representar un subprocesso de un proceso de negocio que se encuentra más arriba que nuestra área de observación. Técnicamente un subprocesso tiene que ser iniciado por un evento técnico, en este caso un evento indefinido.

Eventos intermedios también pueden sobreponerse a actividades y subprocessos para iniciar un tratamiento por excepción, pero como en el nivel descriptivo no modelamos casos de excepción desistimos de esta posibilidad.

El Gateway basado en eventos (Event Gateway) lo excluimos por las mismas razones mencionadas anteriormente: La reacción ante diferentes sucesos muestra una lógica operativa que no nos interesa modelar en el nivel descriptivo.

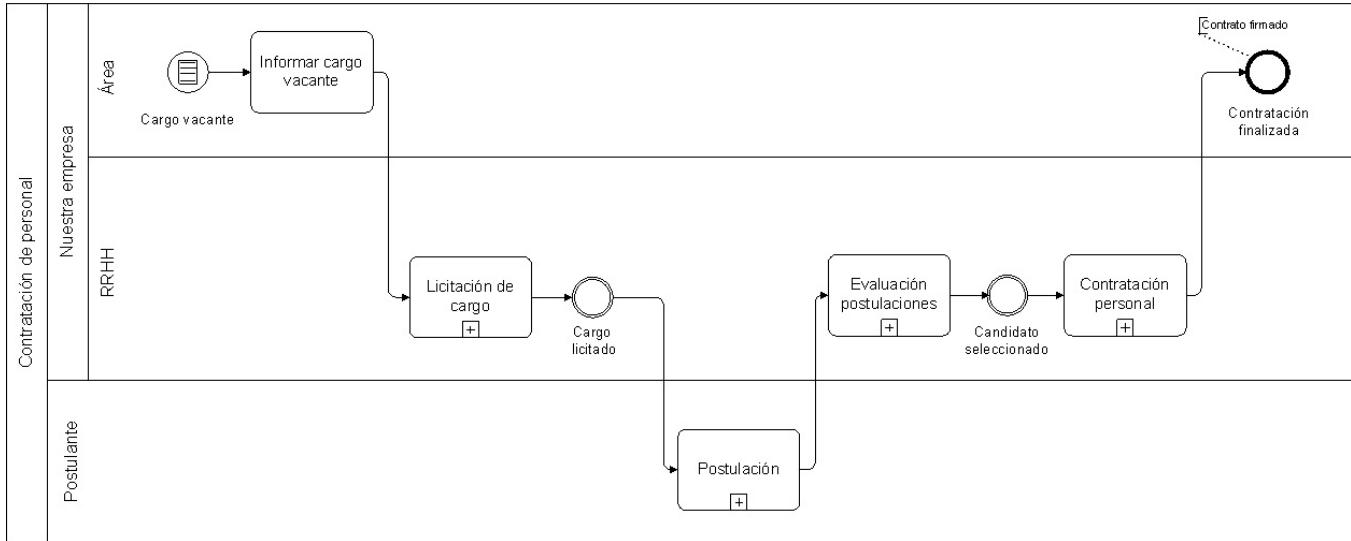


Figura 3.10: Definición de hitos en el proceso de contratación de personal

3.3.5. Datos y artefactos en el nivel descriptivo

El comentario de texto está permitido en el nivel descriptivo y nosotros lo utilizamos bastante seguido. En nuestro ejemplo del proceso de contratación de personal, nos ayuda a explicar que el evento de término está acondicionado a la existencia de un contrato firmado.

También el objeto de agrupación apoya bastante en dar explicaciones a relaciones que sólo se ven en la lógica operativa, pero por lo general los modelos en el nivel descriptivo son tan agregados que no requieren de mayor explicación. No le vaya a suceder a usted como lo hemos visto en algunos casos que confunda el artefacto de agrupación con el objeto que muestra un subprocesso expandido.

La utilización de los objetos de datos puede rápidamente sobrecargar los diagramas del nivel descriptivo, pero otro lado si uno los utiliza con mesura pueden servir para mostrar:

El input y el output del objeto de negocio que se va transformando durante el proceso de negocio.

El tipo de comunicación entre los participantes del proceso.

El segundo punto representa en realidad una colaboración a través del flujo de mensajes entre los participantes de diferentes pools en un proceso, pero como desistimos de esta representación en el nivel descriptivo, podemos utilizar en reemplazo los objetos de datos para mostrar una interacción que sea importante.

Ante la pregunta a Roberto de como fluye la información en el proceso de contratación de personal, explica: «Como he dicho anteriormente, recibimos el aviso via e-mail. La licitación de cargo la ponemos en varios medios, en nuestra página

web y en diferentes bolsas de trabajo en Internet. Tradicionalmente recibimos las postulaciones por correo, pero en el último tiempo cada vez más vía e-mail.»

Los diferentes canales de cómo fluye la información, los podemos modelar con objetos de datos, los que asociamos al flujo de secuencia entre las actividades (ver figura 3.11). El objeto de input lo asociamos, por lo general, al flujo de secuencia entre el evento de inicio y la primera actividad del proceso y el de output entre la última actividad y el evento de término. Formalmente esta representación no es del todo correcta, porque el output no se entrega al evento de término, pero para representar el caso de negocio en el primer nivel es aceptable.

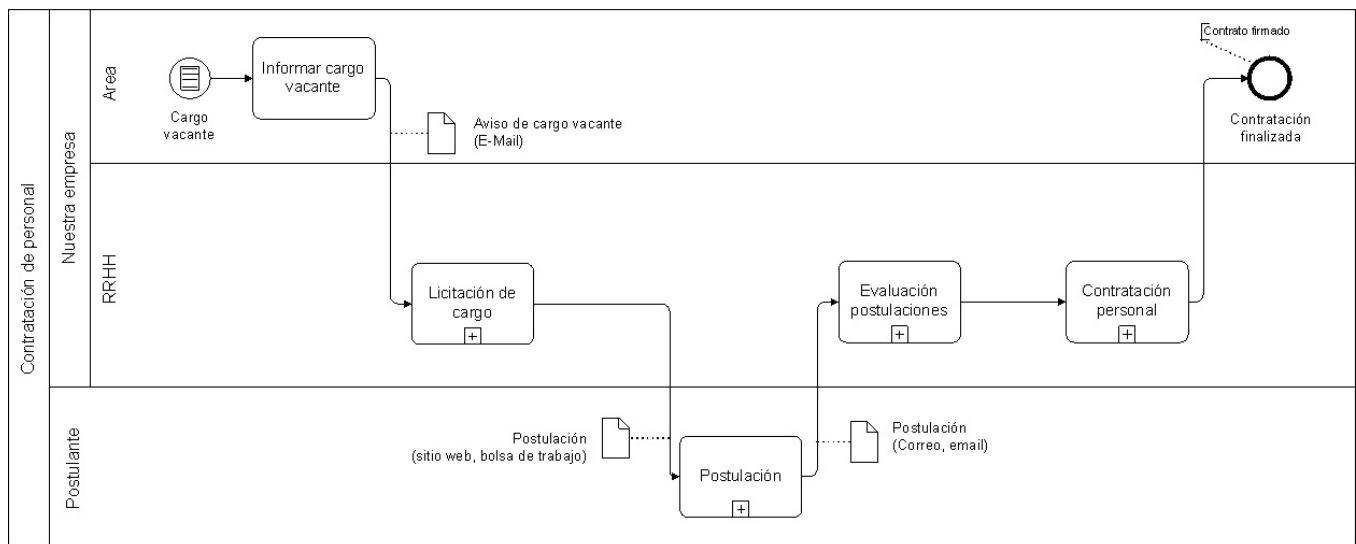


Figura 3.11: Modelo de proceso de contratación de personal con objetos de datos en nivel descriptivo

3.3.6. Artefactos propios en el nivel descriptivo

En el capítulo 2 le explicamos que BPMN permite utilizar simbología propia en calidad de artefactos. Los artefactos sólo se pueden asociar a objetos de flujo (actividades, Gateways, eventos) para que no afecten el flujo de secuencia. Su función es darle más expresividad a los diagramas sin influir en el flujo de control.

De acuerdo a nuestra experiencia, los artefactos propios se prestan bastante bien en el nivel descriptivo para mostrar ciertos aspectos que queremos retener en el proceso. Por lo general se quiere mostrar el apoyo de aplicaciones de negocio en las diferentes etapas del proceso. Aplicaciones se representan por símbolos como tamborcitos o un pc como puesto de trabajo.

Ante la pregunta a Roberto sobre qué sistemas de apoyo utilizan para administrar el proceso, responde: «Hasta el momento casi ninguno. El perfil del cargo se describe en una tabla de excel, todo el resto del proceso lo hacemos sin apoyo de ningún software.» La figura 3.12 muestra este caso.

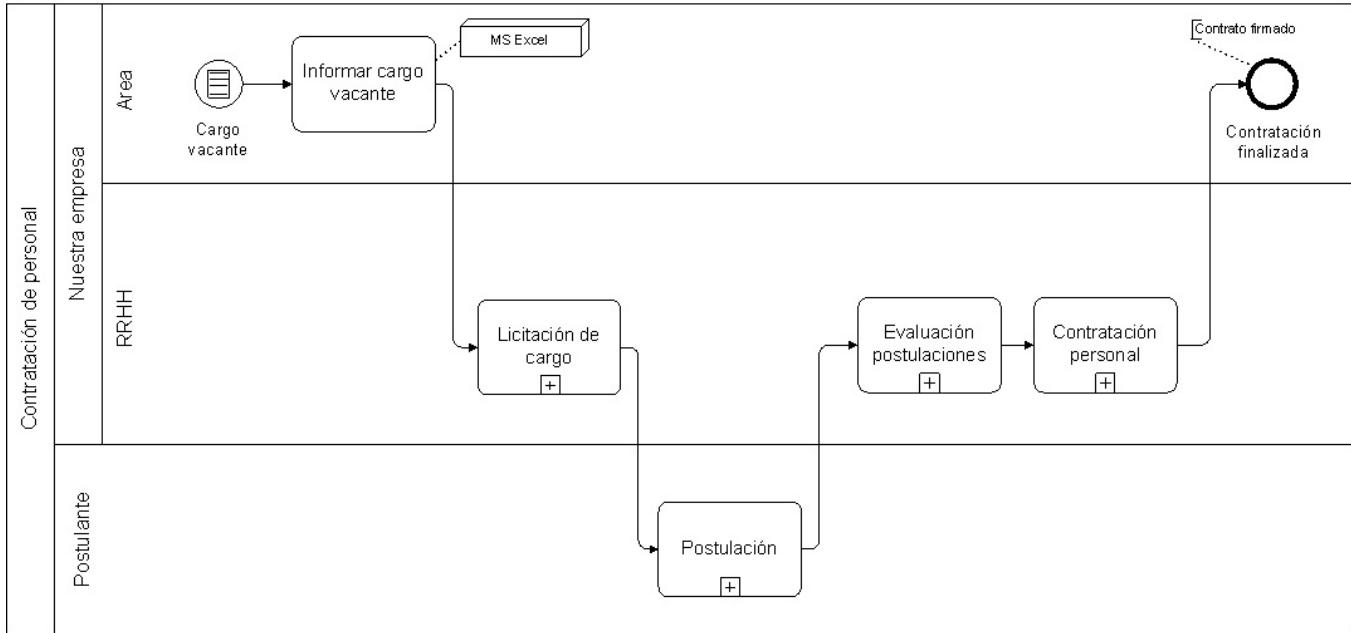


Figura 3.12: Modelo de proceso de contratación de personal con artefactos propios en nivel descriptivo

Cada rubro tiene sus propias particularidades y esto lo podemos representar en BPMN con artefactos propios: símbolos de riesgo en el rubro de las aseguradoras, ambulancias para un servicio de urgencia, etc..

3.3.7. Ocultar y mostrar objetos en el diagrama

Nuestro ejemplo completo del nivel descriptivo considerando elementos descriptivos como hitos, flujo de información (con objetos de datos) y apoyo de TI (artefactos) lo encuentra en la figura 3.13.

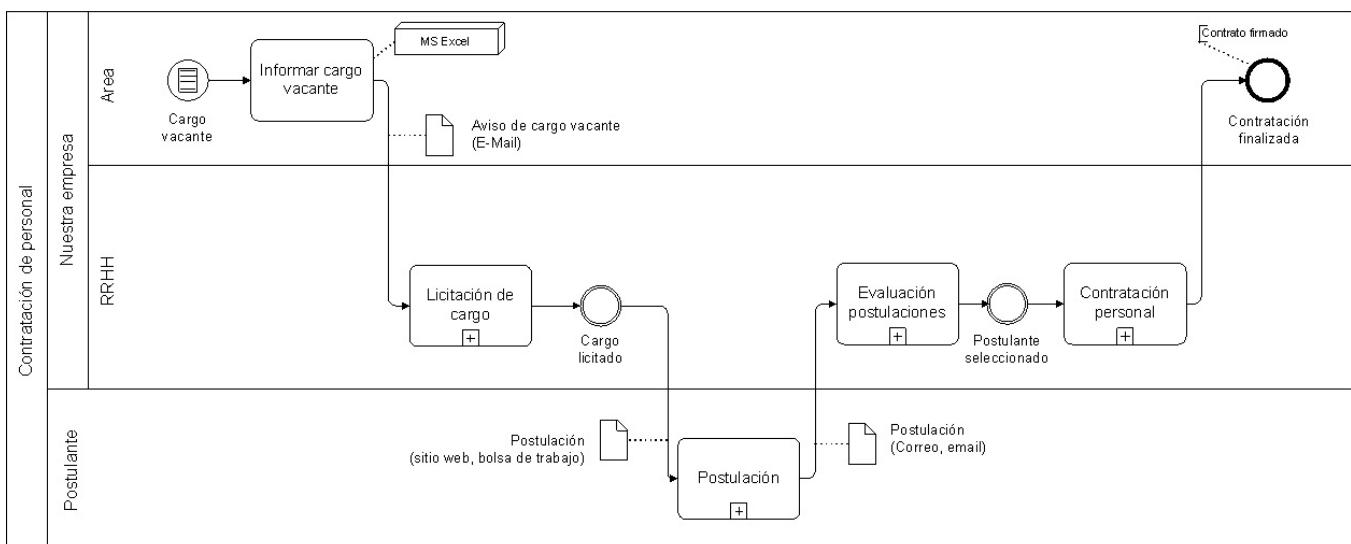


Figura 3.13: El proceso de contratación de personal como ejemplo de nivel descriptivo

La pregunta que surge a continuación, es si es necesario de ver siempre estos

artefactos. Con bastante certeza, podemos afirmar que no es necesario tener siempre estos artefactos a la vista, pero esto no es un asunto de la notación sino de las capacidades que tenga una herramienta de poder filtrar estos objetos. Muchos alumnos nos preguntan a menudo si existen herramientas que ofrecen estas funcionalidades. Al respecto le podemos responder lo siguiente:

El mostrar u ocultar objetos no es propio de la notación, esto lo tiene que ofrecer una herramienta de modelamiento.

Desde el punto de vista técnico es relativamente sencillo desarrollar una funcionalidad que puedas filtrar artefactos y algunas herramientas lo ofrecen.

Pero en el caso de objetos de flujo, propios de la notación o en el caso de los eventos indefinidos que están asociados a los flujos de secuencia, no es tan sencillo de ofrecer funciones de filtro. Si queremos filtrar eventos intermedios que se encuentran asociados al flujo de secuencia, la herramienta tendría que reconstruir el flujo, es decir borrar los que están asociados a los eventos e implementar nuevos sin estos eventos, además tendrían que reposicionarse todos los objetos del diagrama.

Peor aun sería el caso si tenemos después de un evento un XOR-Gateway, entonces la herramienta tendría que calcular cómo reemplazar los flujos de secuencia que fueron borrados junto al evento, un algoritmo muy complejo que no se justifica para el beneficio que podría tener. Por esta razón difícilmente encontrará un proveedor de software que realice o haya realizado esta funcionalidad.

3.4. Reflexiones sobre el Análisis en el Primer Nivel

Luego de haber levantado y documentado el proceso de contratación de personal en el primer nivel podemos continuar el análisis de dos maneras:

Comenzamos con un levantamiento en detalle de la situación actual del proceso y pasamos al segundo nivel de descomposición, o

entramos a una etapa de análisis de mejora a partir de la documentación levantada.

Esto va a depender del objetivo del proyecto. Si queremos certificarnos en las normas ISO o queremos elaborar un manual de procedimiento para los participantes del proceso, tenemos que modelar el detalle de la lógica operativa (segundo nivel).

En nuestro ejemplo, tenemos la misión por parte de Roberto de hacerle propuestas de mejora a su proceso de contratación de personal. Él ya manifestó su descontento con la situación actual, aunque presume donde y porque existen ineficiencias, no tiene fundamentos para identificarlas claramente. Si vemos nuestro ciclo de BPM podemos entrar ahora en la fase de análisis con el objetivo de investigar las causas de los problemas y a partir de ahí hacer propuestas de mejora (ver figura 3.14). Por otro lado, podríamos optar por analizar el detalle de toda la lógica operacional de la situación actual, pero si está claro que se desea una mejora sería un esfuerzo

tremendo levantar la situación actual en detalle, por lo que en práctica por general desistimos de ello y hacemos el análisis con la información levantada en el nivel descriptivo.

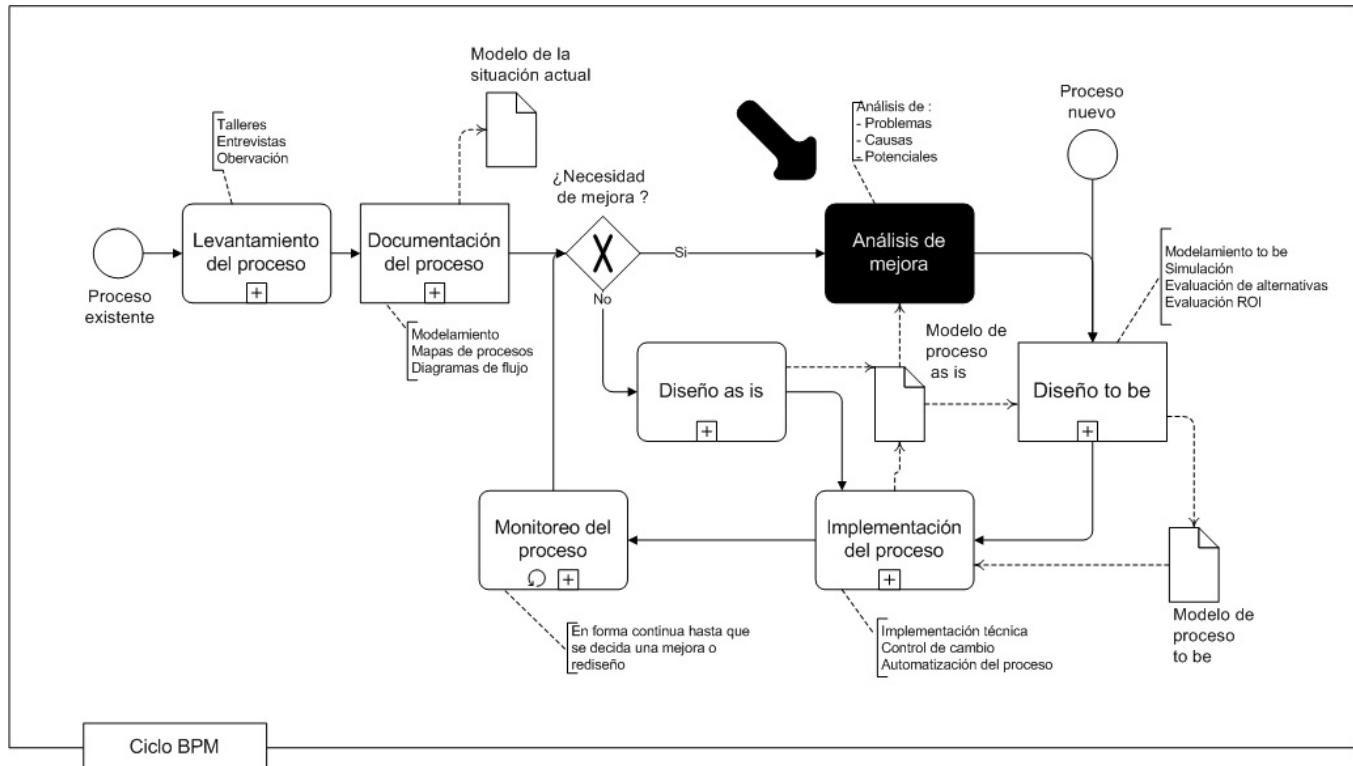


Figura 3.14: La fase de análisis en el ciclo de BPM

¿Cómo iniciamos el análisis? Lo primero que tenemos que aprender es a «escuchar». Tenemos que entrevistar no sólo al «Process Manager», sino también al cliente y a los participantes (usuarios) del proceso. Por lo general es suficiente de realizar uno a dos reuniones con los participantes para identificar los problemas del proceso. Ahora, no es objetivo de este libro de explicarles todas las técnicas de moderación que puede emplear para llevar a cabo estos talleres, pero vamos a simular en forma simple uno de estos talleres aplicado a nuestro ejemplo.

Agendamos un Workshop con el tema «Análisis de potenciales para el proceso de contratación de personal». Los participantes del Workshop son: Process Manager: Roberto

Cliente: Falko, Gerente de Ventas y al mismo tiempo representante de las otras áreas de la empresa

Process participants: Marina, Cristian y Esteban, usuarios de RRHH
Analyst: ¡Usted!

Luego de la usual fase de Warming Up, en el que usted da a conocer la agenda y el objetivo del Workshop, proyecta el modelo del proceso que levantamos (figura 3.1.3) y espera que los participantes se familiaricen con él. Ahora usted puede emplear una serie de técnicas de moderación para entrar a la fase de análisis.

Nosotros nos decidimos por la técnica más sencilla, en la cual enumeramos los principales problemas que fueron nombrados en nuestras sesiones de levantamiento. Como moderador, usted valida por cada uno de los participantes la lista de los supuestos problemas expuestos:

El ciclo del proceso es muy largo.

El proceso es burocrático.

El proceso es poco transparente (poco claro).

Falko se queja principalmente de la larga duración que tiene el proceso, mientras que Roberto reclama la ineficiencia en los diferentes pasos que son necesarios para llevar a cabo el proceso. Ambos coinciden, que se requiere de una mayor transparencia para mejorar la eficiencia del proceso y de esta forma poder intervenir con mayor claridad en cada uno de los pasos.

Este es el momento de entrar ahora en el análisis, de investigar las causas de los problemas y lógicamente también de profundizar en el sentido de ir más allá e investigar las causas de las causas. Algunas de las causas las podemos relacionar directamente a los subprocesos del modelo y otras están más relacionadas con el proceso en general.

La discusión lleva rápidamente a la conclusión que existen muchas actividades manuales que entorpecen el proceso y lo hacen lento. Si pudiéramos automatizar gran parte del proceso, podríamos hacerlo más expedited y más claro. A Falko no le agrada mucho el reclamo de Roberto que las áreas no entregan la descripción de los perfiles de cargo en forma correcta y consistente, pero tiene que reconocer que habitualmente las áreas tienen que iterar varias veces hasta ponerse de acuerdo sobre la descripción de cargo que deberá ser licitada. Falko atribuye la causa de este problema al formulario de la tabla de excel que tienen que llenar; lo encuentra muy engorroso, complicado y sin una pauta para llenar cada uno de los campos en ella. Al respecto dice : «Ni siquiera sabemos cuales campos son obligatorios y cuales son opcionales.»

Un punto más difícil de investigar, es el problema de los largos tiempos de espera en el proceso, es decir el tiempo desde la asignación de una tarea hasta que sea realmente ejecutaba. Sobre este punto se reprochan Roberto y Falko mutuamente sobre la poca disponibilidad de recursos que tienen cada uno de ellos, pero sin que ninguno de ellos pueda fundamentar su punto de vista. Usted como moderador tiene que en este momento detener la discusión, argumentando que efectivamente los tiempos de espera son largos en este proceso, pero que aún no podemos resolverlo sin entrar a un análisis más en detalle al respecto.

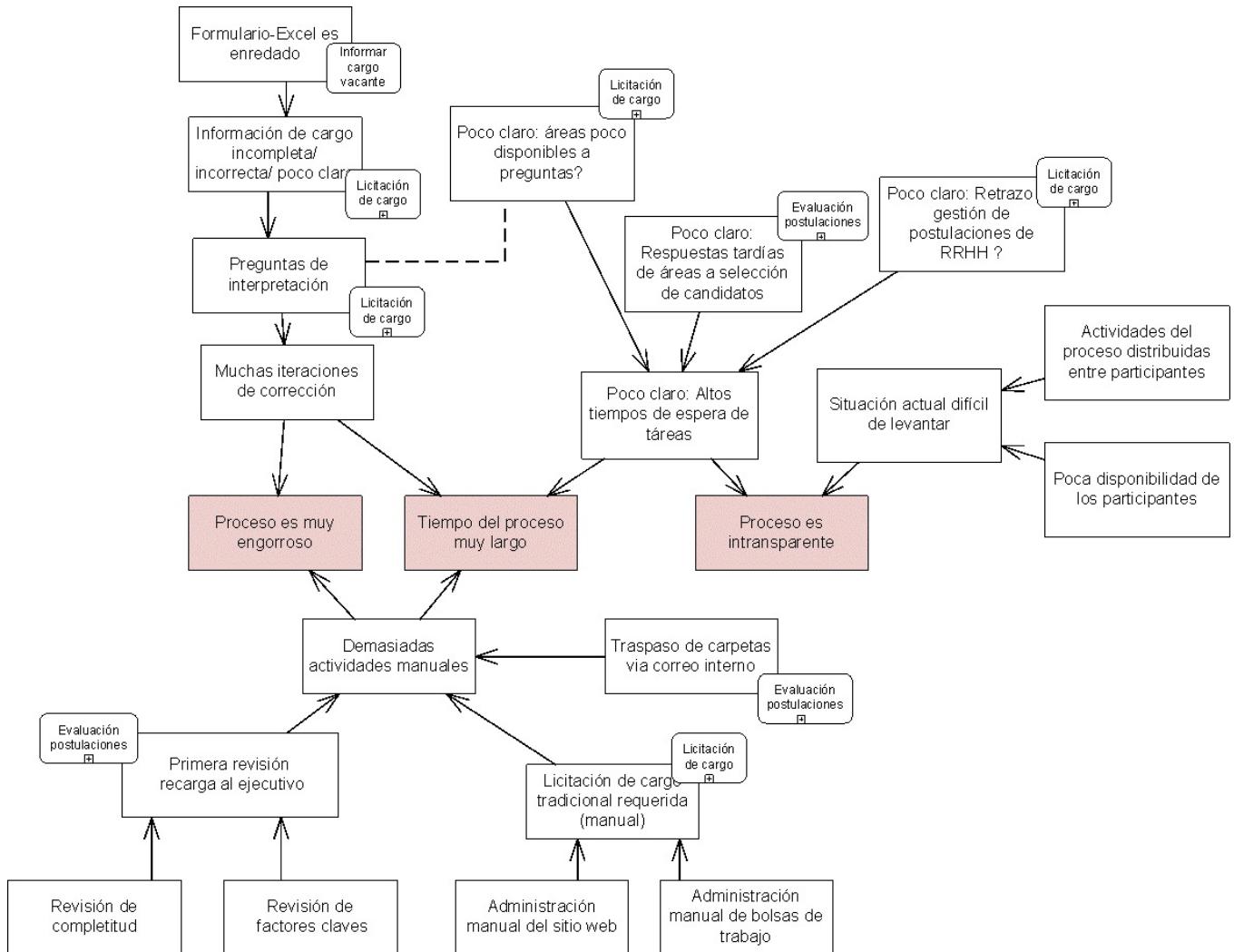


Figura 3.15: Cadena causal como resultado del taller de análisis

El resultado del análisis «causa-efecto» lo puede apreciar en la figura 3.15. A partir de los tres problemas principales del proceso, se identificaron cuatro áreas de análisis que deben ser abordados en el marco de un proyecto de rediseño:

La cantidad de actividades manuales deben ser reducidas.

Deben reducirse la cantidad de iteraciones de coordinación entre áreas y RRHH.

Poder monitorear en línea el estado en donde se encuentra el proceso. Los tiempos de espera deben registrarse y conocerse la asignación de los recursos encargados.

Seguramente, el lector ya está adivinando que la mayoría de estos problemas los vamos a solucionar con apoyo de TI, pero esto no es siempre así. En muchas ocasiones la causa del problema se encuentra en la organización del trabajo, en la asignación de los recursos, en la insuficiente asignación de responsabilidades, u otras causas no nombradas que no tienen solución a través del apoyo de un software. Sin embargo BPMN a través de la automatización de los procesos, puede hacer sin duda los procesos más eficientes, más transparentes y con mejor control. Ese es también el objetivo principal detrás del BPMN y es ese el potencial de mejora que queremos abordar en este caso de estudio y por ende en este libro.

En el ciclo de BPM pasamos ahora a la fase de diseño «to be» en el cual queremos diseñar un proceso que cumpla con los requerimientos formulados en la enumeración anterior. Este es el momento de pasar del nivel descriptivo al nivel operacional.

3.5. Consideraciones: Primer Nivel y BPMN 2.0

La ampliación de la simbología en BPMN 2.0 está orientada fundamentalmente a mejorar la expresividad en el nivel operativo y el nivel técnico, razón por la cual no impacta directamente en el nivel descriptivo de los procesos.

Es difícil de prever la relevancia que puedan tener los nuevos tipos de diagrama de coreografía y de conversación en el nivel descriptivo. Por un lado fueron introducidos para mejorar la visibilidad y la coordinación entre los diversos participantes de procesos, lo que tendría que apoyar el modelamiento en el primer nivel, pero por otro lado se introducen nuevos objetos (símbolos) lo que aumenta la complejidad en el primer nivel. El aumento de complejidad podría actuar en contra de una mejor agregación de los diagramas convencionales.

Si aplicamos estos dos tipos de diagramas a nuestro ejemplo del proceso de contratación podemos apreciar las siguientes variantes como muestran las figuras 3.16 y 3.17. En el primer diagrama del tipo de conversación vemos a primera vista que en el proceso interactúan (conversan) tres participantes. Al postulante le agregamos la marca de «múltiple» para indicar a que a una licitación pueden postular más de un candidato. Al mismo tiempo podemos agregar a las asociaciones la cardinalidad (Nota: Este tipo de diagrama se parece mucho a un modelo de datos), pero por lo general habrá que explicarle al usuario el significado de esta simbología. El diagrama de conversación puede ser de gran utilidad si en el proceso interactúan muchos participantes, entonces podemos reducir el proceso a una representación muy compacta para mostrar la colaboración entre ellos.

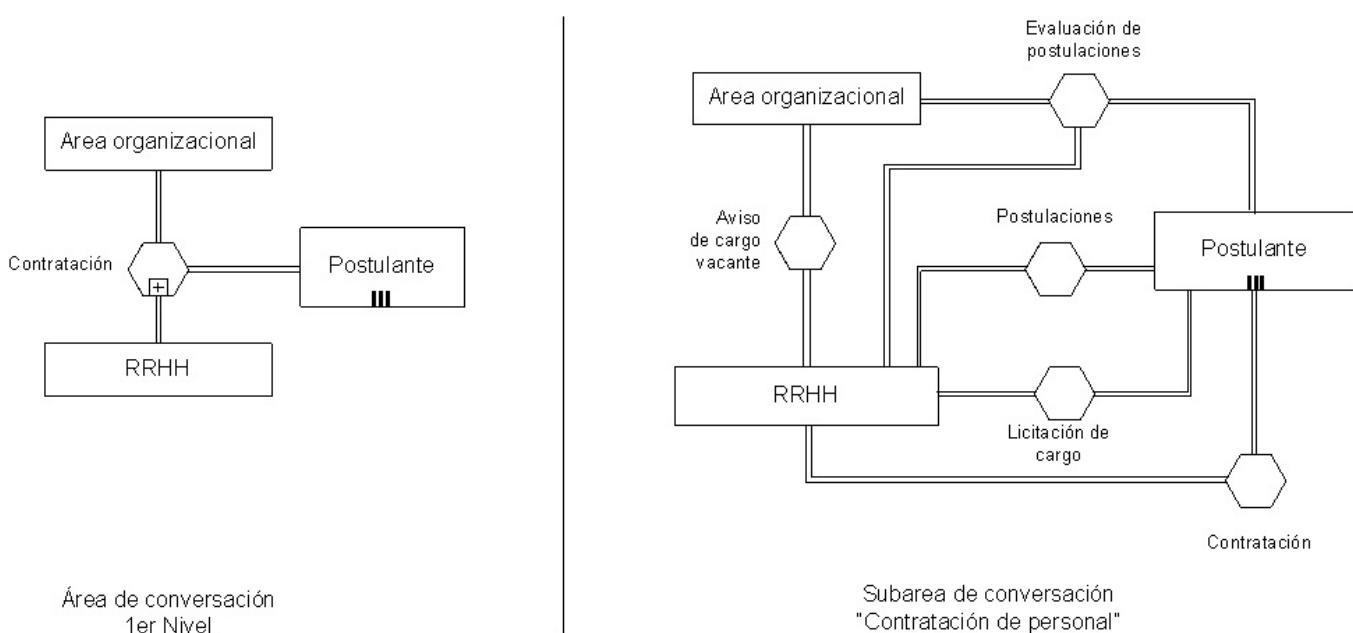


Figura 3.16: Diagrama de conversación en dos niveles de descomposición para proceso de contratación de personal

El diagrama del tipo de conversación también ofrece la posibilidad de desagregarlo y mostrar la colaboración con mayor detalle. El signo más (+) en el hexágono en el diagrama agregado (diagrama izquierdo de la figura 3.1.6) indica que existe un diagrama debajo con mayor nivel de descomposición. En el diagrama desagregado vemos que no todos los participantes conversan entre ellos. El área no participa en la subárea de licitación, ni de postulaciones.

Lo que no se ve en el diagrama de conversación es el flujo, por ejemplo que el postulante envía (flujo de mensaje) una postulación a RRHH, sólo se ve el tema de la comunicación o conversación. Dicho de otra forma, este tipo de diagrama no es un diagrama de flujo, por consiguiente no es un modelo de proceso, pero puede servir para mostrar en forma compacta como se comunican los participantes. En el primer nivel puede servir para mostrar una vista adicional al flujo que aún no ha separado los pools, con el objetivo como lo habíamos visto antes de reducir la complejidad y ahorrarse de presentar a los ejecutivos un diagrama de colaboración en que se muestran los flujos y la interacción por medio de pools separados.

El diagrama de coreografía (ver figura 3.1.7) es más preciso que el de conversación, porque muestra adicionalmente el flujo (dependencias del tiempo en el flujo de comunicación). El diagrama de coreografía es una mezcla entre el diagrama de conversación y el de procesos, porque vemos el flujo y los diferentes participantes en las actividades (subprocesos) del tipo de coreografía. Una ventaja de este diagrama es la combinación de cardinalidad y flujo: La licitación de cargo se realiza sólo una vez y le envía mensajes a los diferentes postulantes. Desde el punto de vista semántico, esta representación no es correcta, porque en realidad no envía los mensajes, pero se entiende. En el siguiente paso los postulantes envían sus postulaciones a RRHH. La denominación del subprocesso de coreografía «Postulación» está bien, porque cada postulante envía sólo una postulación a RRHH. En el caso de «evaluación de postulaciones», también está correcta la denominación plural, porque se evalúan todas las postulaciones. Aquí tenemos entonces, cardinalidad múltiple en «Evaluación de postulaciones» y simple en «Postulante», porque cada postulante envía sólo una postulación. La última actividad de coreografía «Contratación de personal» es singular porque sólo el candidato seleccionado firma el contrato de trabajo.

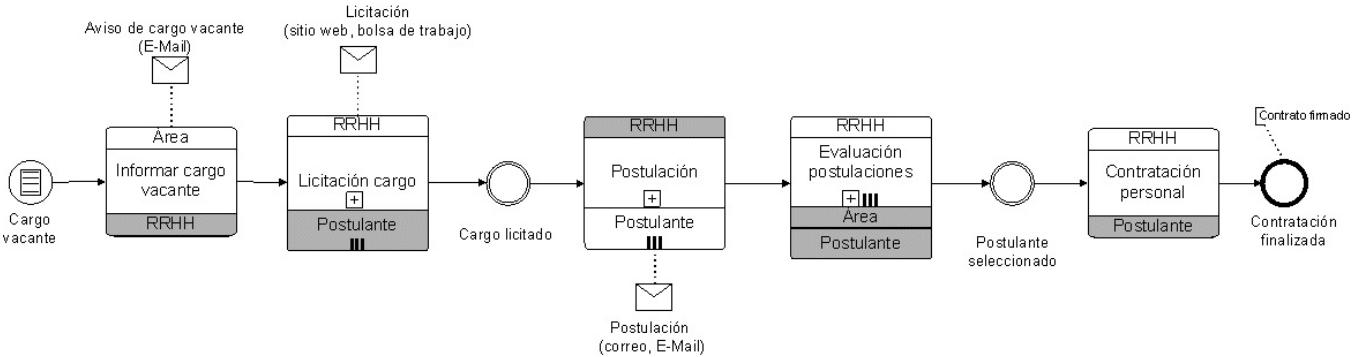


Figura 3.17: Diagrama de coreografía para proceso de contratación de personal

La gran ventaja que vemos en los diagramas de coreografía es la forma compacta con que se pueden representar procesos de muchos participantes que son muy intensivos en el intercambio de información (comunicación). Pueden ser de gran utilidad para acordar protocolos de comunicación y su respectiva lógica en la integración de procesos independientes, como por ejemplo, Business to Business (B2B), Business to Consumer (B2C), o Business to Government (B2G), pero tenemos dudas si este tipo de diagrama sirven para la audiencia del nivel descriptivo. Nunca está demás probar para ver el grado de aceptación por parte de la dirección estratégica.

En procesos de negocio complejos pueden servir al analista como mapas de procesos para entrar al segundo nivel, el nivel operacional, materia que trataremos en nuestro próximo capítulo.

Capítulo 4

Nivel 2: Modelo de Procesos Operacional

4.1. Sobre este Nivel 4.1.1. Objetivo y Beneficios

El modelo operacional, el segundo nivel, es la esencia del BPMN-Framework. Este nivel abarca toda la lógica de negocio en detalle, incluyendo los casos de excepción, identificando las reglas de negocio, y la interacción en detalle con todos los participantes. El modelo operativo le sirve al usuario de negocio (Process Participants) como guía o manual de procedimiento en su trabajo diario. Al analista de proceso (Process Analyst) le sirve como input para evaluar la eficiencia del proceso y poder desarrollar propuestas de mejora. Por último, el modelo operativo constituye la base y el punto de partida para el diseño de una implementación técnica por medio de TI.

Debido a estas razones, la lógica de negocio se describe mucho más detallada que en el nivel uno, lo que constituye un desafío mucho mayor debido a la complejidad que tiene este nivel. El proceso de negocio es por lo general un complejo intercambio de actividades entre personas y sistemas.

El analista de procesos tiene que comprender cómo se realiza el trabajo en su máximo detalle y como interactúa el flujo de trabajo con los sistemas y los usuarios de negocio. Su habilidad de análisis tiene que además identificar puntos críticos y reconocer problemas actuales del flujo de trabajo para hacer propuestas de mejora, tanto técnicas como organizacionales. Su pregunta esencial es:

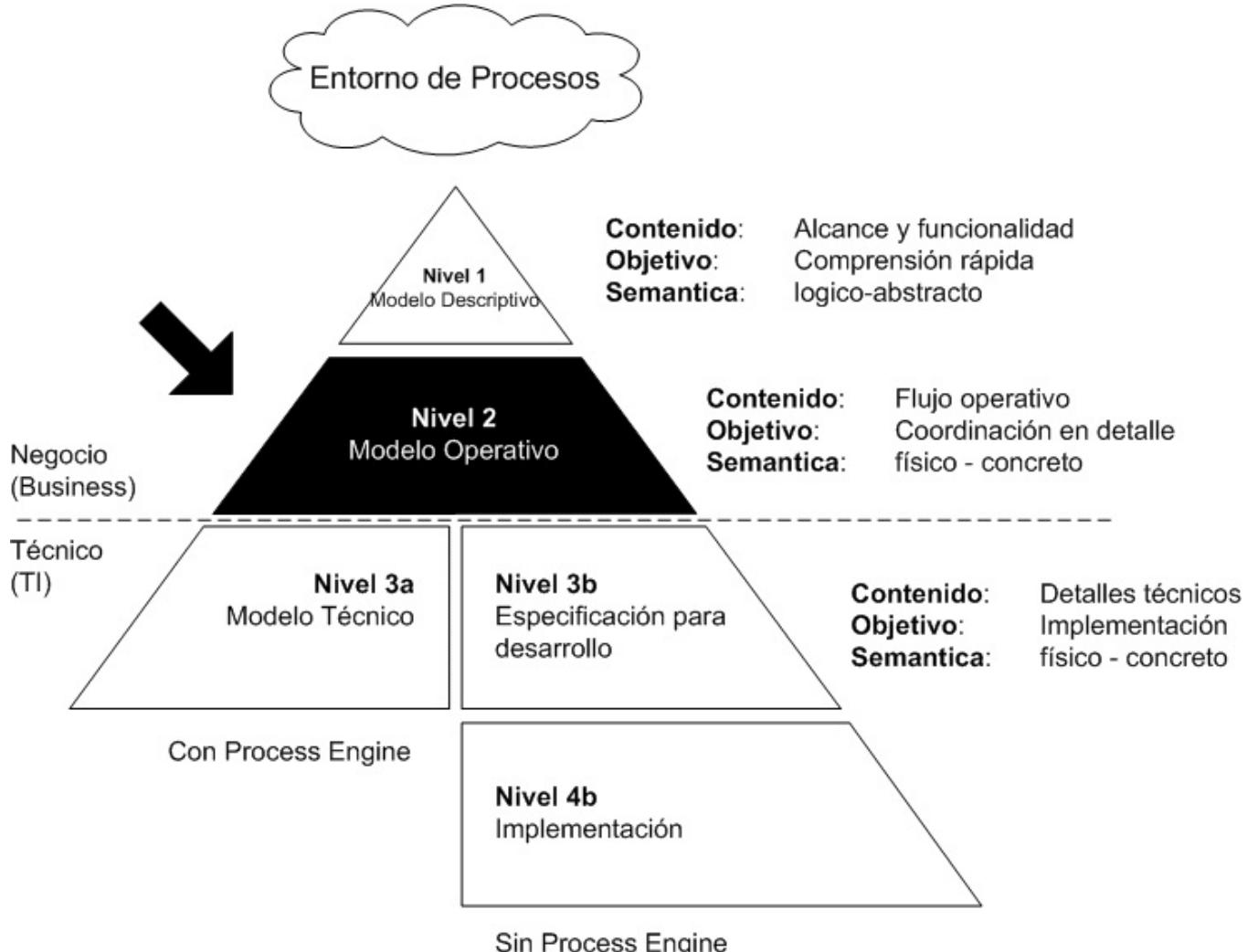


Figura 4.1: Nivel 2 Operativo del Marco Estructural para BPMN

¿Cómo se trabaja y cómo podría hacerse mejor? El usuario de negocio en cambio sólo se interesa en aquellos aspectos del proceso que le conciernen a él. Su pregunta esencial es:

¿Cómo tengo que trabajar? Llegado el momento de querer implementar técnicamente el proceso, entra en juego el ingeniero de procesos (Process Engineer). El ingeniero de procesos no es parte integrante del equipo para el modelamiento en el segundo nivel, su rol activo comienza en el tercer nivel, pero él debe enterarse sobre los requerimientos funcionales que deberán implementarse. Su pregunta esencial en este nivel es:

¿Qué debe cumplir el sistema de TI? Uno de los desafíos principales en el nivel operativo es de coordinar satisfactoriamente estos tres roles principales contestando a cada una de sus preguntas. Nada de sencillo, pero si lo logra, obtendrá una serie de beneficios:

El modelo operativo corresponde en su lógica principal completamente a la lógica que será implementada técnicamente. Logramos un buen alineamiento entre la capa de negocio y la capa de TI. La documentación elaborada no se deja de lado y se

empieza de nuevo a modelar como sucede en muchos proyectos.

La brecha entre la capa de negocio y de TI se reduce considerablemente, se logra una plataforma de entendimiento común: La gente de negocio y la de TI conversan y se coordinan sobre el mismo modelo. Se reconoce de inmediato el impacto que puedan tener requerimientos funcionales o el impacto que pueda tener al negocio una cierta forma de implementación técnica.

En el momento que se implemente la solución, se pueden medir los indicadores de ciclo y agregar otros indicadores de negocio que permitan monitorear en línea el comportamiento de los procesos y generar reportes estadísticos en forma automática.

Podemos resumir: Si usted logró desarrollar los modelos de acuerdo a esta forma de proceder, habrá conseguido «por fin» un medio para alinear la capa de negocio con la de TI, por lo menos en lo que se refiere a modelamiento e implementación de procesos.

4.1.2. Requerimientos para el modelo

Los diagramas de proceso tienen que estar en nivel 2 sintácticamente correctos igual que en el nivel 1, pero adicionalmente al nivel 1 su semántica debe ser consistente. En el nivel 2 se describe como realmente se trabaja, razón por la cual no se pueden aceptar contradicciones o faltas formales, como lo admitimos en el nivel 1.

Si además el objetivo del proyecto es llevarlo a una implementación técnica, debemos considerar adicionalmente los siguientes requerimientos: Todas las preguntas que el ingeniero de procesos le pueda hacer a los usuarios de negocio, tienen que aclararse en este nivel. El nivel 2 tiene que tener la precisión necesaria para llevarlo a un diseño técnico en el nivel 3, sin que tenga que ponerse en duda como se modeló la lógica de negocio. El nivel 3, idealmente, sólo debiera ser un proyecto técnico.

El grado de precisión no es sólo para el ingeniero de procesos importante. El usuario de negocio necesita una descripción detallada para responder a preguntas en sus operaciones diarias. Justamente en este punto nos encontramos con un nuevo desafío desde el punto de vista de como modelar para lograr el entendimiento y alineamiento. Si le mostramos al usuario el modelo completo, habrán partes que no corresponden a su dominio, por lo tanto él tendrá que identificarse dentro del modelo, lo que puede dificultar la lectura y la compresión de este. El analista de negocio y el ingeniero de procesos deben tener una visión general sobre el proceso (respecto al ciclo de BPM), no así el usuario, él quiere ver sólo su área de responsabilidad, para él BPM y la organización del proceso es sólo un medio para la finalidad de hacer bien las cosas.

4.1.3. Procedimiento

De acuerdo a los requerimientos descritos anteriormente, el modelo de procesos en el nivel 2 tiene que ser lo suficientemente preciso, pero tampoco debe llevarse a un grado de complejidad tan alto que se torne inentendible. El requerimiento suena paradojal, porque una descripción detallada y precisa, sea en forma verbal o por medio de diagramas, lleva en la práctica automáticamente a mayor complejidad. Entonces ¿cómo salimos de esta encrucijada? Sólo reconocemos una posibilidad para solucionar este problema: «A cada rol le asignamos una vista propia del proceso completo».

Perspectiva	Process Participant	Process Analyst	Process Engineer
Pregunta clave	„Como tengo que trabajar?”	„Como se trabaja?”	„Que debe hacer el sistema (BPMS)?”
Vista	Orquestación propia	Colaboración	Orquestación del sistema (BPMS)

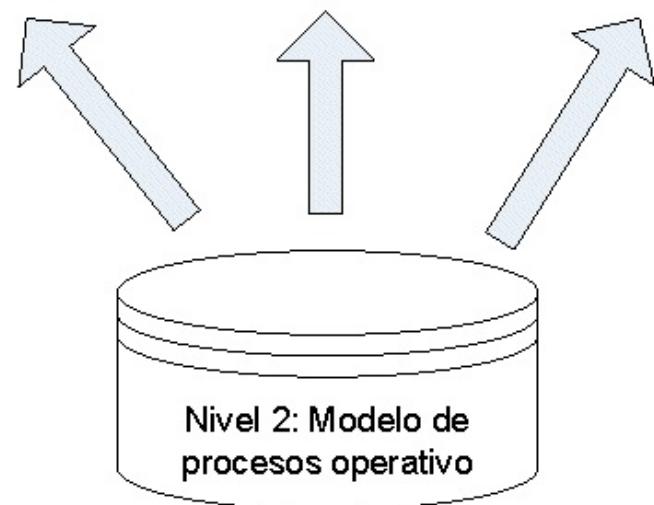


Figura 4.2: Roles y vistas de los participantes en un proyecto de BPM

La figura 4.2 muestra las vistas de los tres participantes clave en un modelamiento de procesos, pero lo que a nosotros nos interesa es sobre todo reducir la complejidad de los roles de negocio, debido a que cada uno tiene su propia responsabilidad sobre sólo una parte del proceso entero. Si a cada tipo de usuario le presentamos o modelamos su propio proceso, es decir sólo visto desde su perspectiva, se va a identificar con él y va a aceptarlo. Él sabe lo que tiene que hacer, cuando es su turno o si tiene que esperar a un evento proveniente de otro lugar. De esta forma, evitamos que el usuario tenga que hacer un filtro y reconocer sus propias tareas. Finalmente la integración de todas estas vistas es lo que hace tan complejos estos diagramas. En

cambio los roles del analista y del ingeniero son técnicos y ellos si requieren ver el «todo» para implementar un proyecto de BPM que es transversal.

El punto esencial de nuestra metodología en el nivel 2 es por consiguiente la clara diferenciación entre orquestación y colaboración, conceptos que explicamos en la sección 2.9: Cada participante recibe su propio pool. Como lo establece la norma, cada pool representa un proceso individual. El diagrama de colaboración, que muestra la interacción entre los pools, pasa a ser entonces un asunto de los analistas, de quienes sí esperamos que tengan dominio sobre la complejidad. En nuestra forma de proceder, el sistema de workflow (process engine) se declara como un participante más en el proceso, es decir le asignamos un pool propio. Este es entonces el pool que debe implementar el ingeniero de procesos. Si seguimos este modelo de diseño en el nivel operativo, estamos cumpliendo claramente con el objetivo que persigue el estándar BPMN, que tiene en su foco el sistema de workflow, con la diferencia que nosotros pensamos en traspasar este principio al workflow que queda afuera de la implementación de TI, y que es el workflow manual o humano.

Esta diferenciación se hace también necesaria, porque en la mayoría de los casos, sólo una parte de todo el proceso de negocio se implementa o puede implementarse técnicamente. Por lo general existen algunas instancias que deben intervenir o tomar decisiones los seres humanos. Si queremos representar el proceso completo, entonces tenemos que levantar lo completo y no sólo reducirlo a la parte TI. Justamente es eso lo que proponemos y le asignamos un pool a cada participante, independiente si se trata de un sistema o una persona.

¿Cómo logramos diseñar un modelo de proceso con estas vistas diferenciadas? El rol principal que debe juntar este puzzle es el analista de procesos. Él tiene que entender a cabalidad BPMN y tiene que ser capaz de modelar el proceso desde la perspectiva de uno de los participantes. Él tiene que levantar y administrar el modelo del proceso desde el nivel 1 y llevarlo hasta la entrega del nivel 3, nivel en donde se realiza el diseño técnico.

Esto puede suceder aplicando los siguientes pasos:

1. Definición del alcance y del modelo «to be» en el nivel 1. La forma de proceder la explicamos en el capítulo 3 de este libro.
2. Separación de los lanes en pools (ver sección 4.2 a continuación). 3. Modelamiento del proceso «to be» desde la perspectiva de cada uno de los participantes. Esto debe hacerse con el dueño del proceso (Process Manager) y los usuarios de negocio (Process Participants). Este tema lo trataremos en la sección 4.3 más adelante.
4. Modelamiento de aquellas actividades que van a ser apoyadas por el sistema de

workflow. También este diseño tiene que validarse con el dueño del proceso y los demás participantes (sección 4.4).

5. Modelamiento de la lógica en detalle del sistema de workflow. Gran parte de esta lógica la podemos deducir de los procesos de los participantes. Este proceso lo debe modelar el analista de procesos, el ingeniero de procesos o bien ambos en conjunto. El modelo aún es independiente de la implementación técnica, es decir aún no es ejecutable, pero puede ser complementado en el nivel 3 por el ingeniero de procesos (sección 4.4.2).

6. Desarrollo y documentación del resto de los requerimientos, como pantallas, datos y reglas de negocio. Estos requerimientos se van agrupando a los objetos del modelo de procesos del sistema de workflow, por medio de marcas y símbolos adicionales del BPMN reservados para estos fines (sección 4.4.3).

Una vez que se haya desarrollado el modelo completo, se pueden validar con los participantes sus vistas del modelo de acuerdo a los roles que se han asignado. Como se habrá dado cuenta el lector, para administrar e ir integrando los niveles, vistas y objetos técnicos es necesario contar con el apoyo de una buena herramienta. Sobre todo que ofrezca la funcionalidad de filtrar, esconder y mostrar, pools, artefactos, etc..

4.2. Del Nivel 1 al Nivel 2

Volvemos a retomar ahora nuestro ejemplo del proceso de contratación de personal que levantamos y modelamos en el capítulo 3, sección 3.2 y como punto de partida de descomposición al nivel 2 lo mostramos nuevamente en la figura 4.3. También analizamos en la sección 3.4 las debilidades que cuenta el proceso manual en su situación actual. Constatamos que la organización del proceso en si, no está mal, pero que hay pérdidas de valor al no contar con ningún tipo de apoyo de TI que permita un mejor control y mayor transparencia en la gestión del proceso. Nuestra misión será entonces ahora de modelar este proceso en detalle con toda su lógica operacional. En la primera etapa analizamos el flujo sólo desde el punto de vista organizacional y finalizada esta etapa, analizaremos que forma un sistema de workflow puede mejorar el rendimiento del proceso en general.

4.2. DEL NIVEL 1 AL NIVEL 2 167

En el capítulo anterior vimos que los modelos agregados del nivel 1 aún mantienen inconsistencias semánticas, las cuales no permiten seguir desagregando directamente los modelos al nivel 2. Consideramos absolutamente normal que los modelos del nivel descriptivo se diferencien de la vista de los modelos del nivel operacional. Podemos partir también de la base que los modelos del nivel descriptivo prácticamente no cambian (son muy agregados), sin embargo en el nivel operativo si

sufren frecuentes cambios, razón suficiente para dejar claro que estos modelos no deben diferenciarse en su estructura fundamental de los modelos técnicos.

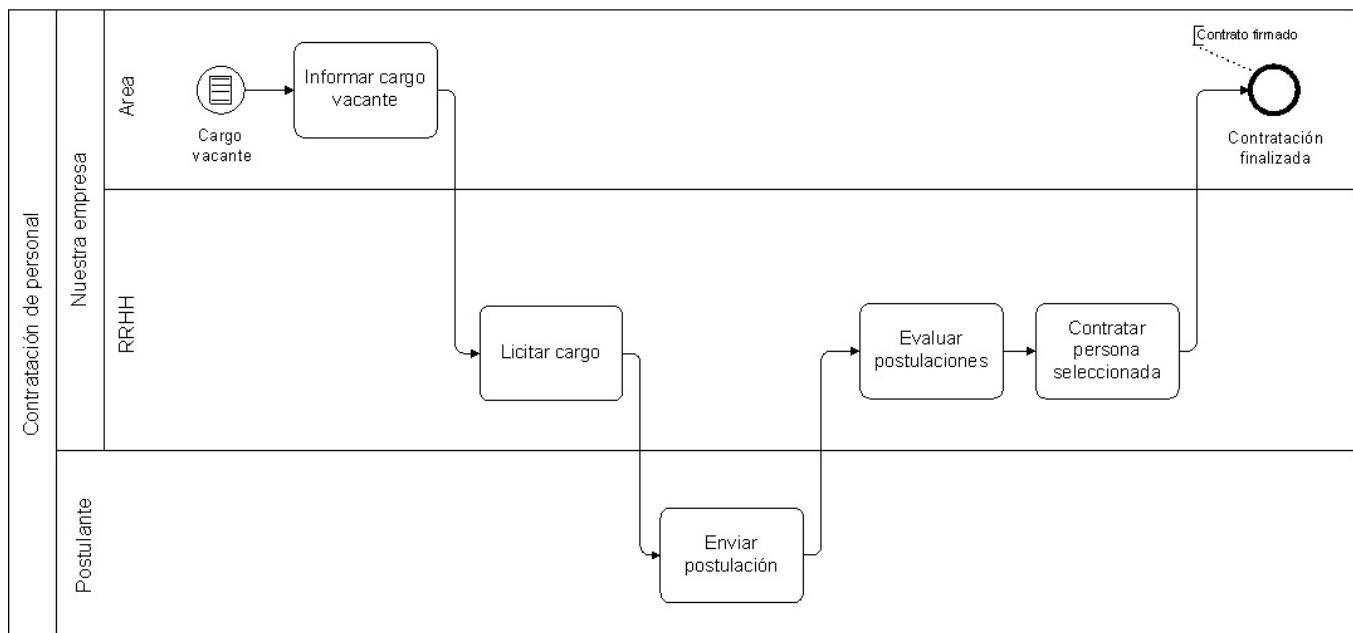


Figura 4.3: Modelo de proceso de contratación de personal para nivel descriptivo

Como consecuencia en la mayoría de los casos, no nos queda otra alternativa, que aparte de obtener un modelo de referencia y orientación, tenemos que modelar nuevamente desde un principio. Esto pareciera ir contra todos los principios de la ingeniería, o una aberración, pero en la práctica no podemos hablar de esfuerzos perdidos. Levantar los modelos del nivel descriptivo no requieren de mucho esfuerzo, una, dos o a lo más tres sesiones cortas. No podemos hablar de un esfuerzo doble, porque hemos definido y validado el alcance, el objetivo principal de ese nivel. Si consideramos que el nivel 1 prácticamente no cambia, no tendremos el problema como del nivel 2 al 3, en donde los cambios si pueden ser frecuentes.

Veamos entonces donde encontramos nuestras primeras inconsistencias en el nivel 1 que nos obligará a repensar el modelo. La primera inconsistencia la encontramos en las diferentes cardinalidades de las instancias que se dan en el flujo del proceso. Si instanciamos un proceso, se da aviso de sólo un cargo que debe licitarse, porque en algunos casos también podrían licitarse para un cargo varias personas, como líneas aéreas que contratan muchos asistentes de vuelo al mismo tiempo, pero pueden postular muchas personas. Todas estas postulaciones deben ser evaluadas una por una. Al final del proceso, sólo una persona es seleccionada y contratada. Otro aspecto importante es que al principio aún no conocemos a los candidatos que van a postular. Todos estos aspectos nos indican que los postulantes deben separarse del pool central y modelarse con un pool propio.

La interacción entre las áreas y RRHH también debe analizarse con mayor detalle en el nivel 2. El nivel 1 muestra como si las áreas no están involucradas en el proceso de licitación y selección, pero en verdad sabemos que existe bastante coordinación en

cada uno de estos pasos como lo analizamos en la sección 3.4. Justamente en esta interacción es donde existe poca claridad y se pierde el seguimiento del proceso, lo que lleva a ineficiencias y largos tiempos de ciclo. Debido a esta razón también separamos estos dos participantes principales en dos pools. De esta forma podemos visualizar y aclarar las interfaces entre ambas unidades organizacionales. Otro argumento válido para separar los procesos de cada unidad es que separamos las vistas de cada uno para no enredar el flujo y poder delimitar claramente las responsabilidades de cada una de estas unidades organizacionales.

En la figura 4.4 apreciamos el proceso de contratación de personal, si separamos las lanes del modelo original del nivel 1, en pools separados por cada uno de los participantes, considerando algunas actividades adicionales que son necesarias para la coordinación entre ellos. El Postulante reacciona ahora a la señal que se licitó un cargo al que quiere postular. Las tres líneas verticales que se ven al medio en el pool del postulante, indican que se trata de una múltiple instancia (ver sección 2.9.5), es decir este pool representa múltiples instancias, razón por la cual modelamos detrás del evento de mensaje un AND-Split. Mostramos en este modelo que RRHH no espera sólo una postulación, sino que procesará cada una de las postulaciones que reciban. El evento terminal al final del pool de RRHH, indica que buscará todas las marcas activas de postulaciones que no han sido revisadas y las finalizará, debido a que se tomó la decisión de ocupar el cargo.

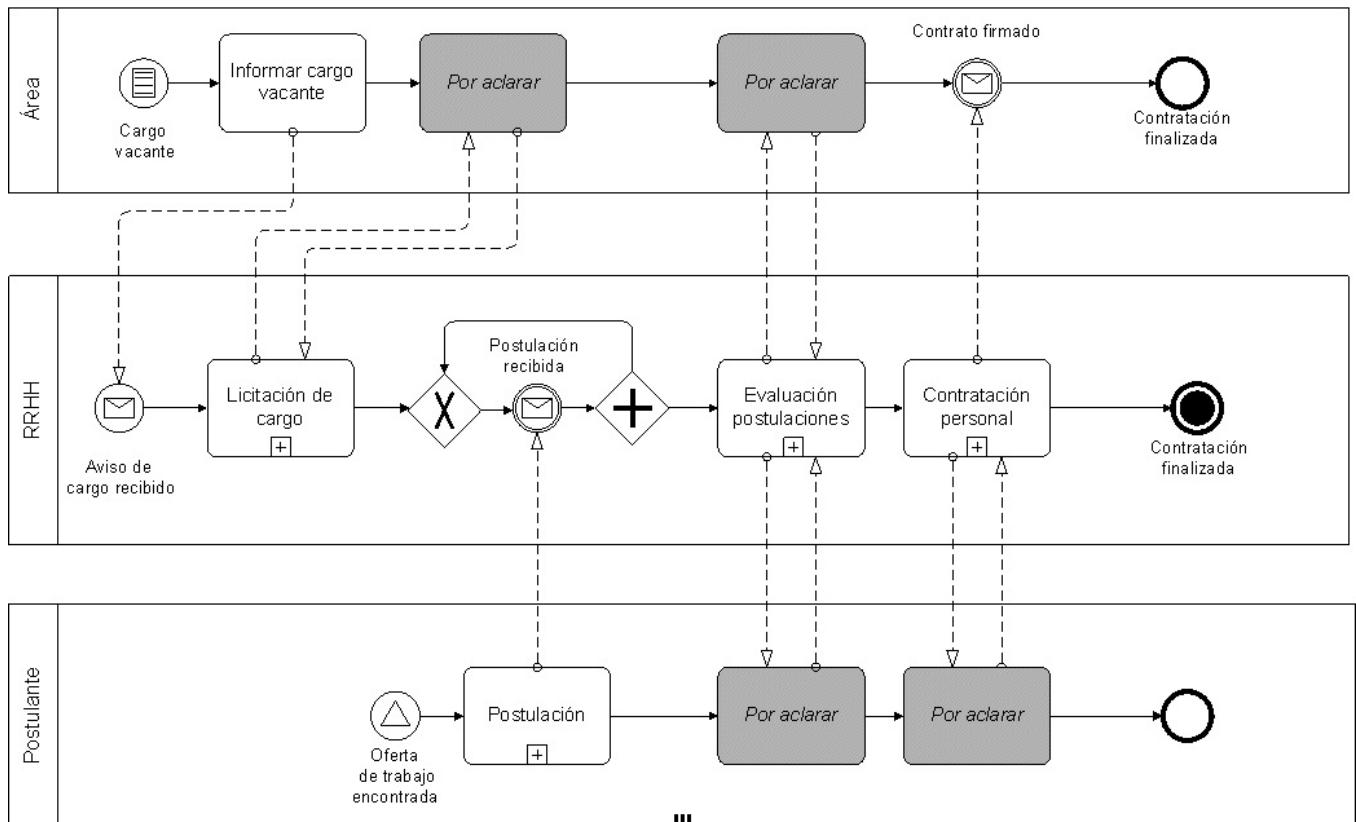


Figura 4.4: Modelo que muestra el comienzo del traspaso al nivel 2

Nos queda por analizar las actividades provisorias, que nombramos «por aclarar» y

de modelar en detalle cada una de las vistas de la unidades organizacionales, es decir aún tenemos bastante por analizar y modelar. También tenemos que investigar los casos excepcionales que se pueden dar. Por ejemplo, que:

Un candidato no clasifique y sea rechazado,
ninguno de los candidatos clasifique,
no se reciban postulaciones a la licitación.

Por razones didácticas no vamos a tratar todos estos casos de excepción y nos vamos a concentrar en modelar en detalle el caso descrito más arriba, desde que se produce una vacante de cargo hasta la licitación de este. Modelar el proceso completo de contratación de personal, para estos efectos sería muy extenso, por lo que nos reducimos a una vista parcial de este proceso de negocio.

4.3. Procesos de los Participantes

El encargado del modelamiento de procesos en el nivel 2, como lo hemos descrito anteriormente, es el analista de procesos. ¿De quién recibe el analista la información detallada para modelar los procesos en este nivel? En la mayoría de los casos de los usuarios de negocio, es decir de la gente que está en el día a día de las operaciones. En nuestro caso de estudio, ahora restringido al subprocesso de licitación, vamos a entrevistar a Falko, jefe del área de ventas que fue nombrado representante del resto de las áreas para este proyecto.

Falko describe su rol como jefe de área para el proceso de licitación de la siguiente forma:

«Cuando necesito contratar a alguien o se desocupa un cargo, doy aviso vía el formulario existente a RRHH. Entonces espero a que RRHH elabore una descripción de cargo para la licitación, la cual reviso y apruebo si corresponde para que sea licitada. Si no está ok, pido hacer correcciones de acuerdo a mis comentarios y vuelvo a esperar para revisarla nuevamente y aprobarla. En algunas ocasiones, RRHH se dirige a mi haciendo preguntas sobre la descripción y el perfil requerido, que lógicamente accedo y entrego la información faltante.»

Si modelamos el proceso que describe Falko e incluimos la interacción con el usuario de RRHH como pool cerrado, obtenemos el diagrama, que aprecia en la figura 4.5.

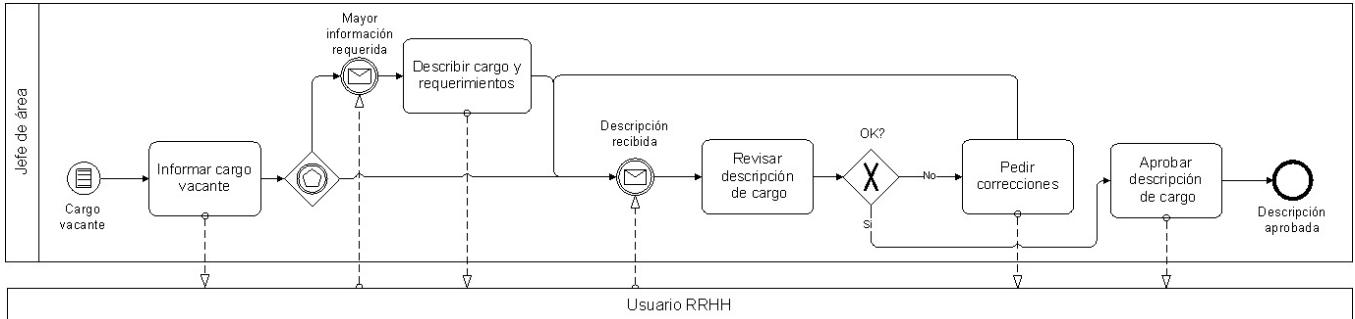


Figura 4.5: Proceso de licitación de cargo desde la perspectiva del participante jefe de área

Comentario: En BPMN 2.0 está prohibido que varios flujos de secuencia entren a un mismo evento intermedio que le siga a un Event-Gateway. Nosotros no le encontramos a esta regla sintáctica algún sentido, pero a pesar de haber insistido en varias ocasiones en la OMG, para que la revisen, no hemos logrado hasta el momento que suceda. Estrictamente deberíamos de modelar este caso como lo muestra la figura 4.6, para quedar 100 % conforme a la regla actual.

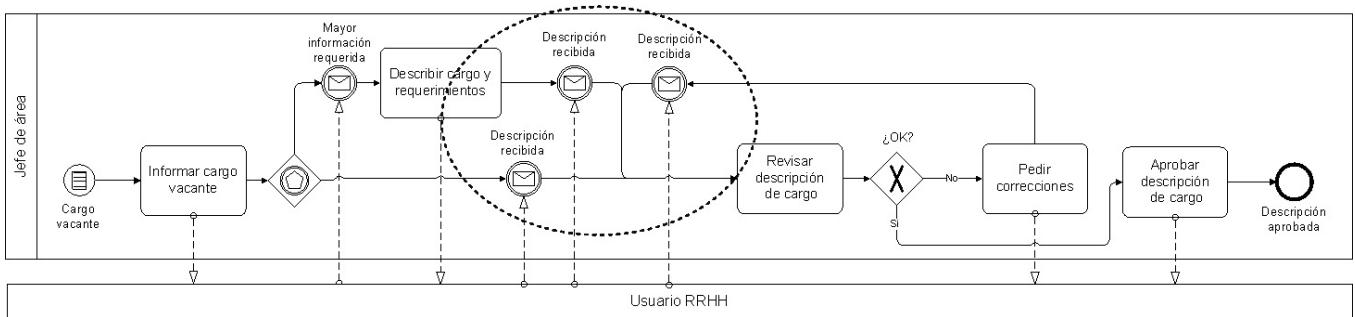


Figura 4.6: Proceso de cicitación de cargo según normas de BPMN 2.0

En el siguiente paso entrevistamos a un usuario de RRHH para que nos describa su perspectiva del proceso de licitación de cargo. Nos ponen a disposición para esta entrevista a Cristian, un usuario con experiencia en RRHH en la empresa:

« Cuando recibo el aviso de un cargo vacante, elaboro una descripción de cargo de acuerdo a la información que se me ha suministrado y tengo disponible. En ocasiones, el formulario está incompleto o surgen dudas sobre la descripción, entonces solicito mayor información al área correspondiente. Luego entrego la descripción de cargo y espero que la aprueben. Puede suceder que el área no la apruebe y solicite correcciones. Entonces hago las correcciones solicitadas y la vuelvo a enviar para su debida aprobación. Una vez que haya sido aprobada la descripción, preparo la licitación.»

El proceso recién descrito lo encuentra modelado en la figura 4.7.

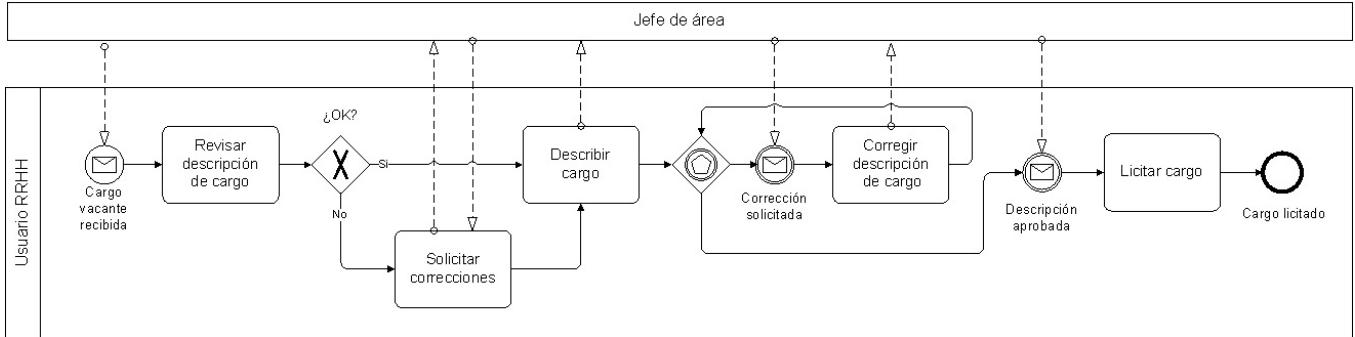


Figura 4.7: Proceso de licitación de cargo desde la perspectiva del participante usuario de RRHH

¿Qué hemos logrado hasta el momento? Levantamos los detalles operativos del proceso de licitación y modelamos dos procesos, es decir cada una de sus vistas por separado. Ninguno de los dos podríamos decir que son complejos, pero con mayor razón nos sirven para efectos didácticos.

Lógicamente que el lector tiene que tener ciertos conocimientos básicos de BPMN para entender estos modelos. Él tiene que entender:

como se interpretan los eventos, sobre todo los eventos intermedios, la diferencia entre Gateways basados en datos y en eventos, la diferencia entre flujos de secuencia y de mensaje.

Podemos deducir entonces que las exigencias para entender modelos del nivel operativo son bastante mayor que para el nivel descriptivo, aún cuando todavía no hemos hablado sobre aspectos técnicos de una solución de software. El nivel 2 es el entorno principal del analista de negocio en el cual él analiza y documenta la lógica operacional en detalle que le va a servir para elaborar una especificación para la implementación tanto técnica como organizacional, como mostraremos más adelante. El dominio del BPMN debe ser una de sus competencias centrales, puesto que él es el responsable de los modelos en este nivel y también porque él mismo elaboró estos modelos. La audiencia a quienes están dirigidos estos modelos son los usuarios de negocio, cuyo flujo de trabajo está representado en estos modelos, como en nuestro ejemplo Cristian y Falko. Los participantes son los usuarios clave para el analista de negocio, razón por la cual estos deben al menos comprender e interpretar estos modelos. Una vez validados estos modelos pueden elaborarse manuales de procedimientos para ellos. Nuevamente nos preguntamos: ¿Serán aceptados estos modelos por los usuarios de negocio?

Nosotros hicimos la experiencia sobre su aceptación y consideramos los siguientes aspectos:

Los participantes ven sólo su propio pool y no los enfrentamos con la complejidad del proceso de negocio completo. Para que este requerimiento se cumpla tenemos

que aplicar estos principios de modelamiento en BPMN y también necesitamos de una herramienta poderosa que nos permita abstraer la complejidad y aislar las vistas de cada pool.

Las participantes fueron capacitados en las estructuras fundamentales de BPMN y cuentan con una guía que les permite consultar aspectos metodológicos en caso que lo requieran. Para estos efectos se recomienda elaborar una guía práctica para los usuarios de BPMN y de la herramienta.

La figura 4.8 muestra el proceso de licitación de cargo como un diagrama de colaboración entre los dos participantes principales, las áreas y RRHH. Queda claro que este diagrama «expandido» es bastante más complejo que si sólo observamos la lógica de un participante y el pool cerrado del otro. Este tipo de diagrama ya no será tan fácil que sea aceptado por los participantes, porque requiere de un mayor sentido de abstracción para que sea entendido, pero no es necesario de confrontar a los usuarios con este nivel de complejidad. El diagrama de colaboración extendida está dirigida al analista de procesos, que gracias a su experticia en BPMN es capaz de entender. En las siguientes dos secciones vamos a hablar sobre los diagramas de colaboración como medio para ir preparando el diseño para la implementación técnica de los procesos.

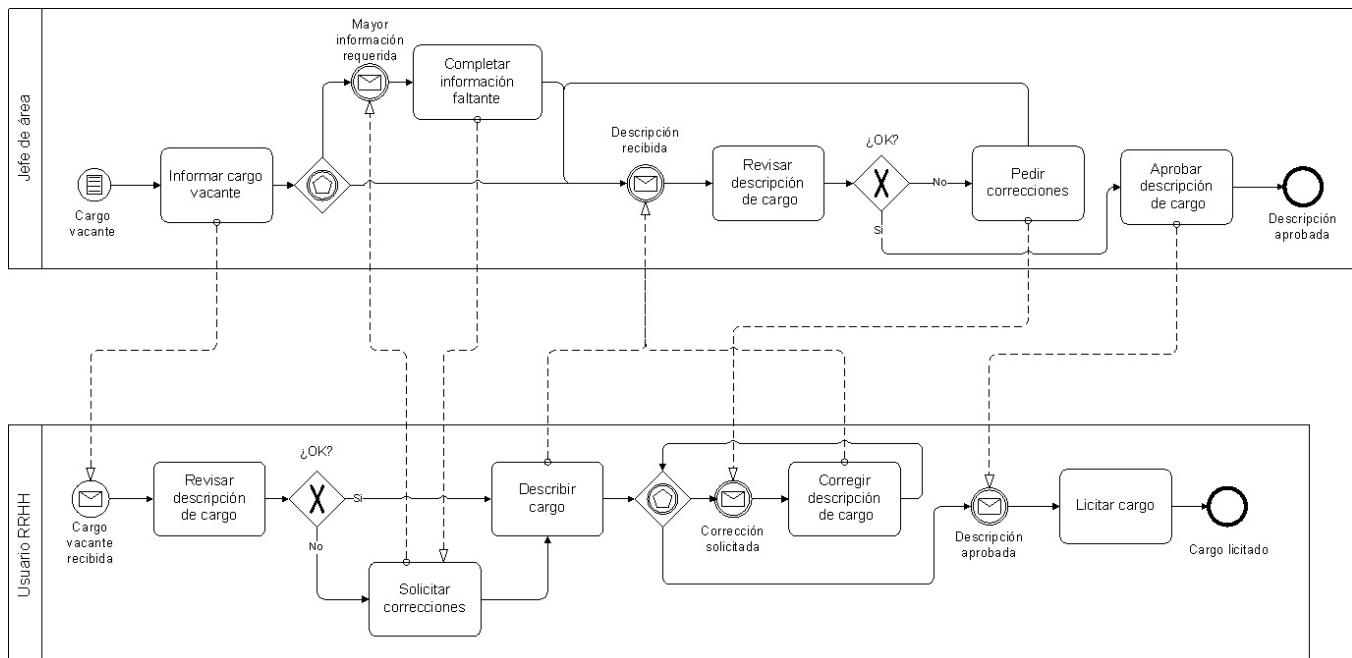


Figura 4.8: Diagrama de colaboración para proceso de licitación de cargo

Con la nueva versión de BPMN 2.0 tenemos la posibilidad de reducir la complejidad de los diagramas de colaboración empleando los diagramas de coreografía (ver figura 4.9). La ventaja que tenemos con este nuevo tipo de diagrama es que podemos representar en forma mucho más compacta la interacción entre los participantes. La desventaja es que no se visualiza la lógica interna, es decir la lógica que no participa en la colaboración. No vemos por ejemplo que el usuario de

RRHH, ejecuta la actividad «Licitar cargo». Por consiguiente el diagrama de coreografía no puede reemplazar el diagrama de colaboración, pero puede ser una ayuda para representar adicionalmente vistas más compactas.

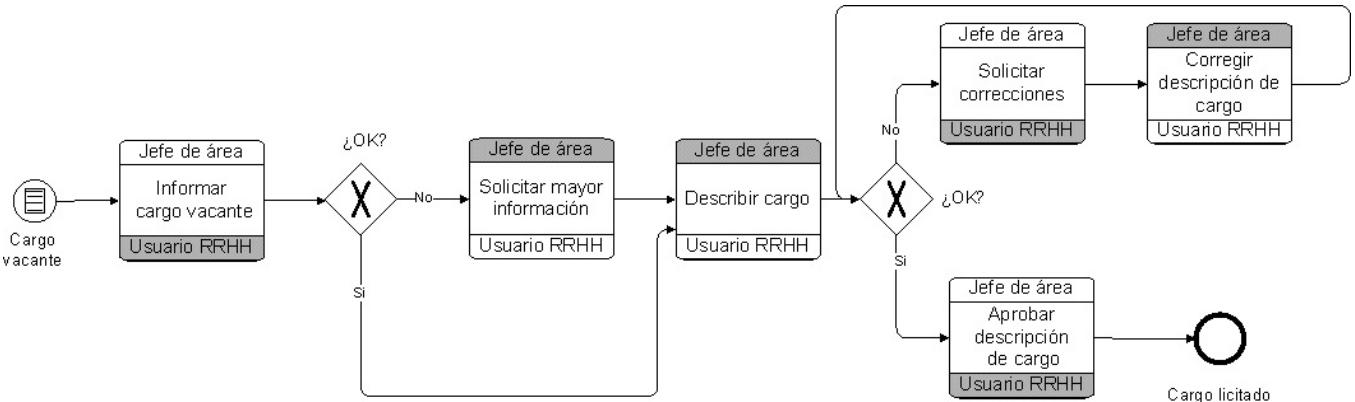


Figura 4.9: Diagrama de coreografía para proceso de licitación de cargo

4.4. Preparación de la Automatización de los Procesos

La descripción de la lógica de negocio es sólo uno de los objetivos en el nivel operativo de nuestro framework de BPMN. El verdadero «Santo Grial» es el traspaso o el mapeo sin variaciones de aquella parte de la lógica de negocio que se quiere implementar por un sistema de workflow, es decir el traspaso del nivel 2 (diseño lógico) al nivel 3 (diseño técnico). En la sección 1.1.4 mostramos como un modelo técnico de procesos es interpretado y ejecutado por un motor de workflow, sistema que combina el «Human Workflow Management» con la orquestación de servicios. Desde el punto de vista de TI es el objetivo principal de cómo la tecnología puede automatizar o implementar BPM, razón por la cual vamos a tratar en las siguientes secciones y sobre todo el capítulo 5 que está destinado a esta finalidad (ver figura 4.10).

Una alternativa, no tan actual y elegante, sería de programar la lógica de los procesos en forma convencional con lenguajes como java, .Net o C#, en vez de utilizar un BPMS o un Process Engine. Este escenario lo vamos a tratar en la sección 4.4.5.

4.4.1. Consideraciones al implementar procesos con un process engine

Volviendo a nuestro caso de estudio, podemos decir que en nuestros talleres con los ejecutivos y usuarios también levantamos el deseo de flujo que se quiere automatizar para mejorar el control y la transparencia de los procesos. Habiendo levantado la lógica de negocio de la situación actual, sin considerar qué partes se van a implementar, estamos ahora en condiciones de tratar este punto. Nuestra misión es ahora ver al participante desde la perspectiva de un usuario de un sistema y aclarar

con ellos, qué requerimientos y funcionalidades debe cumplir la aplicación. Para estos efectos, vamos a introducir un nuevo participante en el proceso de negocio, el sistema de workflow, que va interactuar con los usuarios y compartir mensajes.

Falko, como jefe de área describe la funcionalidad deseada para el proceso de licitación de cargo de la siguiente forma:

«Si se desocupa un cargo lo quisiera ingresar en el formulario estandarizado en nuestro portal, luego lo envío a RRHH. Para revisar la descripción de cargo quisiera recibir una notificación en mi bandeja de entrada del portal y poder revisar esta en la misma aplicación. Si requiere de correcciones la vuelvo a enviar, de lo contrario

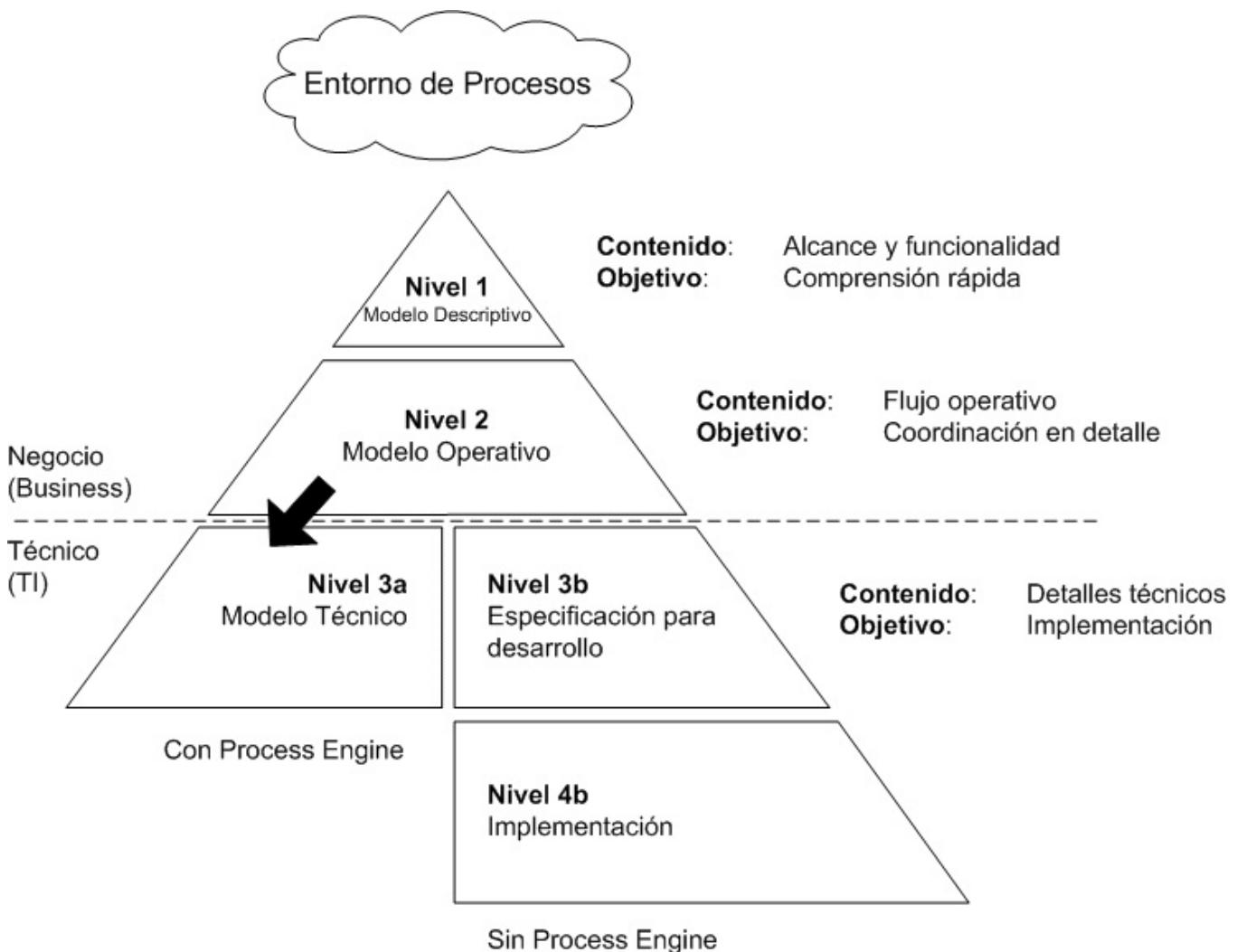


Figura 4.10: Transición del nivel 2 operativo al nivel 3a técnico

la apruebo en el mismo sistema. Una vez que haya sido licitado el cargo quisiera recibir una notificación por e-mail para estar informado que el proceso se inició.» Si se acuerda de la descripción del proceso que nos dio Falko anteriormente (ver figura 4.5), va a encontrar muchas similitudes con ella, pero nos percatamos de dos importantes diferencias:

La interacción con el área de RRHH, no pide que se realice en el portal, sino en

forma tradicional vía e-mail o teléfono.

La notificación que se realizó la licitación no estaba prevista anteriormente.

Tomamos esta descripción del proceso y la ampliamos:

Dividimos el pool del área en dos lanes, «portal» y «otros(canales)».

Asignamos todas las tareas que deben ser ejecutadas por el sistema en el lane «portal». Actividades con flujos de mensajes de salida representan que el usuario terminó su actividad manual (Human Task) y envía el resultado. El XOR-Gateway muestra que el resultado de la actividad puede tener dos estados, «solicitar correcciones» o «aprobar descripción de cargo».

La primera actividad en el lane portal RRHH «informar cargo vacante», no es una actividad que haya sido gatillada por el sistema, debido a que no es capturado por un evento de mensaje. Este evento representa la posibilidad que el mismo usuario inicie el proceso y llene un formulario adjunto que dispone el sistema.

Las actividades de interacción directa con el área de RRHH las asignamos a un lane, que denominamos «otros», porque no son parte del flujo que pasa por el portal, sino que a través de los canales convencionales como e-mail o teléfono. La notificación que la licitación fue realizada también la posicionamos en este lane porque se efectúa vía e-mail. Este e-mail lo envía el sistema, pero el usuario lo recibe en su correo normal y no como mensaje en el portal.

El modelo recién descrito lo encuentra modelado en la figura 4.11. El sistema de workflow se presenta como un participante, pero aún como pool cerrado.

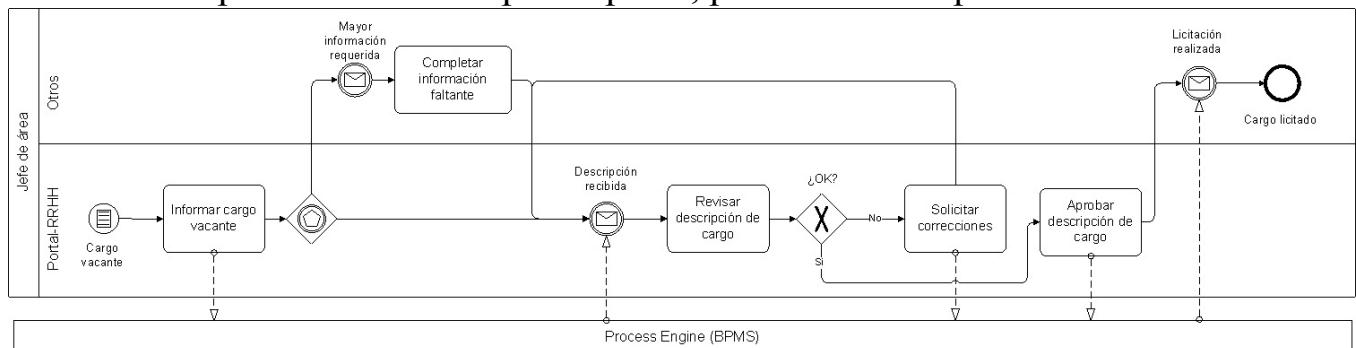


Figura 4.11: Proceso de licitación de cargo apoyado por Process Engine para área de negocio

A su vez el usuario de RRHH describe la funcionalidad que se desea del sistema de la siguiente forma:

«Un aviso de cargo vacante aparece como tarea en mi bandeja de entrada de nuestro portal de RRHH. Al abrir la tarea describo el cargo y luego la envío para revisión al área correspondiente en el mismo sistema. Si tengo que corregir la descripción de cargo lo recibo también como tarea en mi bandeja de entrada en el portal. Sin embargo cuando la descripción fue aprobada, recibo del sistema la asignación una

nueva actividad llamada «seleccionar canales de licitación». Una vez seleccionado los canales paso a la siguiente actividad en donde tengo que tomar la decisión cuando ejecutarla. Una vez que haya sido realizada la licitación, quisiera recibir un e-mail con la notificación que fue efectuada.»

El modelo recién descrito por el usuario de RRHH, lo aprecia en la figura 4.12. A diferencia del modelo de la figura 4.11, el proceso no es gatillado por el usuario de RRHH sino a través de un evento de mensaje, es decir el sistema reacciona ante un evento e inicia el proceso asignándole una tarea al usuario.

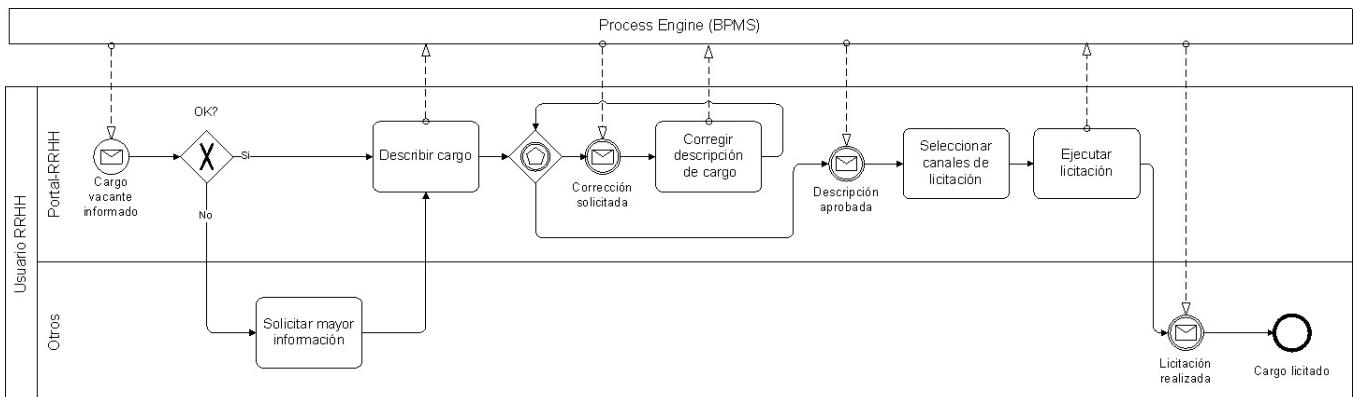


Figura 4.12: Proceso de licitación de cargo apoyado por Process Engine para área de RRHH

4.4.2. El flujo propio del process engine

En este momento nos encontramos en el límite hacia el nivel 3. Como analista de proceso vamos a dirigirnos ahora al ingeniero de procesos (Process Engineer). Con él vamos a aclarar como el sistema puede implementar esta funcionalidad para cumplir con los requisitos nombrados.

Para estos efectos vamos ahora a integrar todos los modelos desarrollados en un gran diagrama de colaboración. Vamos a abrir los pools de los modelos manuales y el pool del sistema de workflow (hasta ahora cerrado). Este último lo vamos a dividir en tres lanes:

Un lane para el jefe de área y uno para el usuario de RRHH. Todas las actividades que se asignan a estos lanes representan, tareas de usuario en el sistema (Human Task).

Un nuevo lane que va representar los pasos automáticos. Se trata de invocación de servicios (Service Tasks), programas internos (scripts), llamados de subprocessos etc..

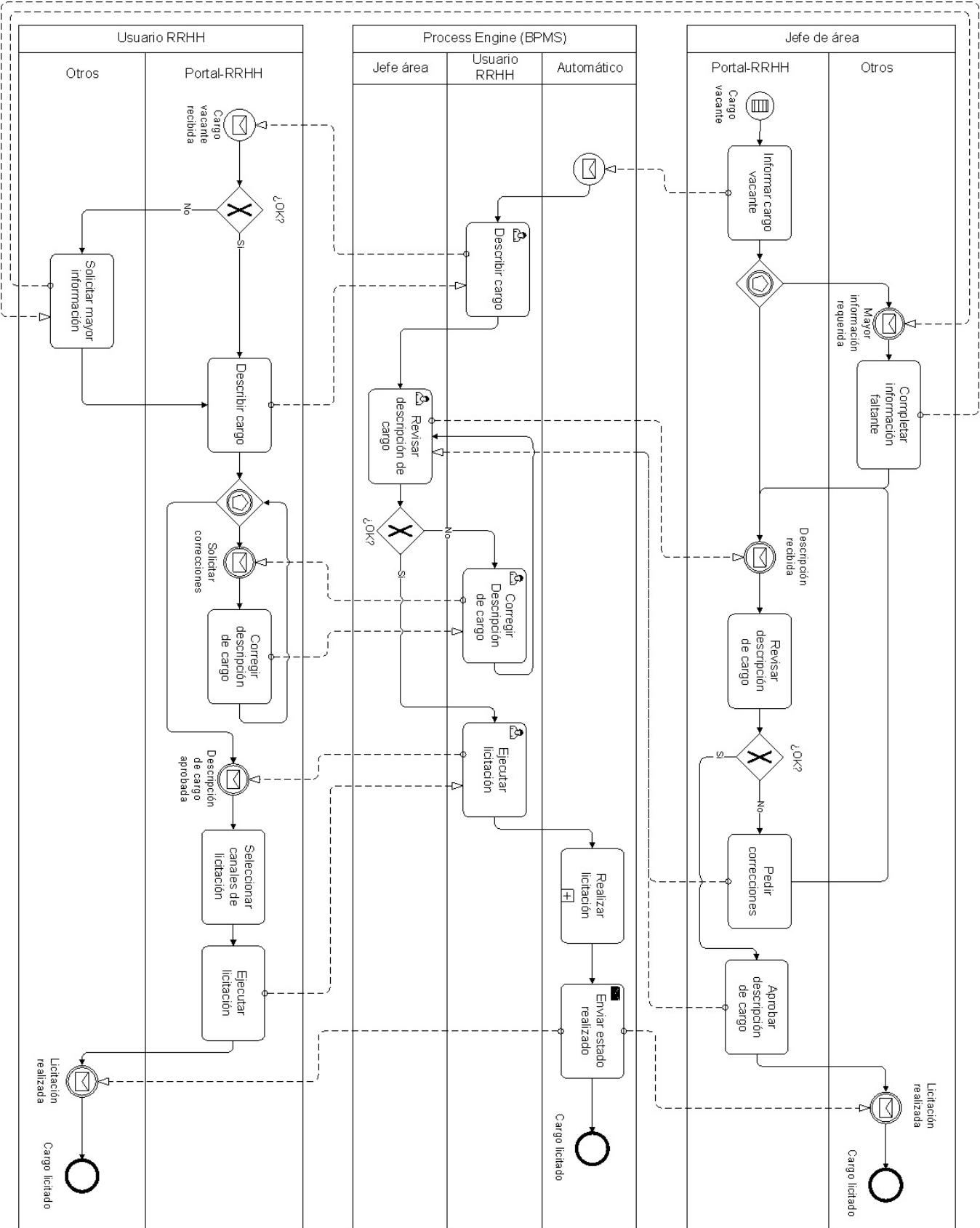


Figura 4.13: Representación del proceso de licitación en un sistema de workflow

A continuación vamos a describir el proceso de licitación apoyado por un sistema de workflow que el lector encontrará en la figura 4.13:
 Falko inicia el proceso cuando informa sobre un cargo vacante al llenar el

formulario en el portal de RRHH y lo envía a través del portal. Este envío lo representamos con un evento tipo mensaje que gatilla el proceso en el sistema e inicia una nueva instancia de proceso. El sistema le asigna en la bandeja de entrada la actividad «describir cargo» a Cristian. Cuando Cristian haya terminado su actividad, el sistema le asigna a Falko la actividad «revisar descripción de cargo». El resultado de esta actividad es la aprobación o la solicitud de «corregir la descripción de cargo». Dependiendo del flujo el sistema va asignar a Cristian la actividad «ejecutar licitación» o «corregir descripción de cargo». Si el flujo es el camino de la corrección, el sistema le va a volver a asignar a Falko la actividad «revisar descripción de cargo», y esto se va repetir tantas veces como Falko no apruebe la descripción de cargo. La siguiente actividad que le asigna el sistema a Cristian es «ejecutar licitación», pero el sistema no lo hará hasta que reciba del portal el mensaje que los canales fueron seleccionados. Cuando Cristian toma la decisión de ejecutar la licitación, el sistema invoca automáticamente el subprocesso «realizar licitación». Este subprocesso invoca una serie de interfaces internas y externas para activar los canales seleccionados. En el último paso el sistema envía en forma automática un e-mail de notificación a los dos participantes, con el mensaje «licitación ejecutada, fecha y hora».

El diagrama de colaboración contiene una cierta redundancia, porque la lógica de los usuarios está representada por un lado en pool propios y además aquellas de estas actividades que fueron implementadas técnicamente (lane en el pool del sistema). Pero al mismo tiempo logramos claramente la separación de roles y responsabilidades en este proceso sobre aquellas actividades que no las controla el sistema de workflow, porque las decisiones se toman por fuera. Por ejemplo decide Cristian, si está en condiciones de hacer la descripción o si no existe claridad llamar a Falko por teléfono para pedir mayor información (esto sucede por fuera, no bajo el control del sistema). Este flujo no lo conoce el sistema, ni siquiera se da cuenta que existe un flujo, pero indudablemente es parte del proceso de negocio, sólo que no está automatizado. Por otro lado tenemos el caso, en este ejemplo, en la cual, el sistema toma el control y decide lo que tiene que hacer Cristian: la lógica de la actividad «revisión de descripción de cargo» está dentro del sistema, es decir la decisión del resultado de esta actividad la toma Falko, pero la lógica de lo que hay que hacer en el caso de aprobación o desaprobación la toma el sistema y no Cristian. Con esta forma de proceder estamos dando metodológicamente solución a un problema que se da mucho en la práctica de modelamiento al intentar de mapear el modelo de la capa de negocio al de tecnología. Casi nunca es implementado «end to end» toda la lógica operacional, de alguna u otra forma casi siempre quedan afuera algunos flujos que siguen bajo el control de los seres humanos. ¿Encontramos el «santo grail»? La mayoría de los analistas de negocio, no distingue cual parte de la lógica «completa» se va a implementar y cual no. Entonces puede ser que obtengamos un modelo correcto en la capa de negocio pero «no» implementable. ¿No le ha pasado que no entiende por qué el proyecto fracasó, si la lógica estaba correcta incluso validada por los usuarios? ¿O por qué en el diseño técnico, los

implementadores botaron a la basura su modelo y empezaron de nuevo? La razón es porque si mezclamos las instancias de control, es decir la humana y el sistema, en un pool, en un diagrama de procesos, no es posible implementar este modelo sin rediseñarlo.

La segunda ventaja de esta metodología es que delineamos perfectamente las responsabilidades de cada uno de los participantes del proceso:

El analista de procesos ve completo el diagrama de colaboración. El implementador o ingeniero de procesos ve sólo el pool o proceso que debe implementar.

Los participantes del proceso ven sólo sus propios pools. Estas vistas individuales no son sólo menos complejas que el diagrama de colaboración, sino que además contienen información adicional que son importantes de captar para comprender el proceso de negocio en general.

Este enfoque, es desde nuestro punto de vista la única forma en BPMN para cerrar las brechas entre la capa de negocio y de TI en el modelamiento de procesos.

4.4.3. Otros requerimientos para la implementación

¿Podría el ingeniero de procesos implementar en TI la solución sólo con la información del diagrama? La respuesta es clara, por supuesto que no. Existen aún muchos requerimientos por aclarar, por ejemplo la interfaz de usuario y los formularios con los campos respectivos de información que se requiere en el diálogo. Estos requerimientos no tienen diferencia alguna con los clásicos proyectos de desarrollo o de configuración de soluciones de software. Estos requerimientos no están directamente relacionados con la lógica de los procesos, pero son necesarios de considerar en la implementación. Nosotros no recomendamos de seguir con el diseño técnico en los diagramas de BPMN, sino hacerlos directamente en la plataforma que va a implementar los procesos. Lógicamente si usted tiene la posibilidad en su entorno de linkear las actividades del diagrama de BPMN, con las actividades del entorno de implementación, tanto mejor. De esta forma el modelo de negocio de BPMN queda integrado a la plataforma técnica. Cualquier cambio debería hacerse siempre primero en el modelo de negocio.

En la tabla que se muestra en la figura 4.14 clasificamos y enumeramos los típicos requerimientos técnicos, que deben considerarse adicionalmente en la etapa de diseño técnico. Si estos requerimientos los documenta directamente en el entorno de implementación o utiliza otros medios depende de la organización de su proyecto.

Tipo	Descripción	Ejemplos	Notaciones	Elemento
Funcional	Funcionalidad que debe cumplir la solución	- Lógica del proceso - Características - Caso de uso - Interfaces - Reglas de negocio	- BPMN - UML (UseCases) - User Stories - Test de aceptación - Texto general	- Actividad
No funcional	Requerimientos técnicos que debe cumplir la solución	- Service Level Agreements (SLA) - Tiempo de respuesta - Robustez - Control de cambio	- Texto	- Pool
Interfaz de usuario	Canales y capa de presentación que interactúa con el usuario	- Pantallas - Workflow guiado - Mobile Devices - E-Mail-roles	- BPMN - Prototipo de pantalla - User Stories - Test de aceptación	- Actividad
Datos	Información que tiene que procesar el software	- Información - Restricciones - Formatos - Canales - Mappings	- Diagramas-ER - Diagramas UML clase - Tablas	- Pool - Objeto de datos
Reglas	Decisiones que tiene que tomar el software	- Validaciones - Revisiones - Cálculos - Puntos de control	- Tablas - Árboles de decisión - Texto	- Actividad

Figura 4.14:

Requerimientos adicionales para la implementación técnica

Siguiendo con nuestro ejemplo del proceso de licitación de cargo, le mostramos como podría implementarse el diálogo de la lógica del proceso del pool del sistema (ver figura 4.15). El lector puede apreciar un prototipo de las pantallas del diálogo y su relación con la lógica del proceso. La lógica de control está integrada en las opciones de las pantallas, por ejemplo Falko puede dar el «ok» o desaprobar la descripción de cargo (opción segunda pantalla arriba).

4.4.4. Implementación de puntos de integración

Dependiendo del entorno de implementación que haya seleccionado para su organización, se hará necesario integrar componentes a su plataforma técnica para implementar procesos. Específicamente dependiendo de la arquitectura de software disponible, tenemos que aclarar como se va a implementar:

Las reglas de negocio,

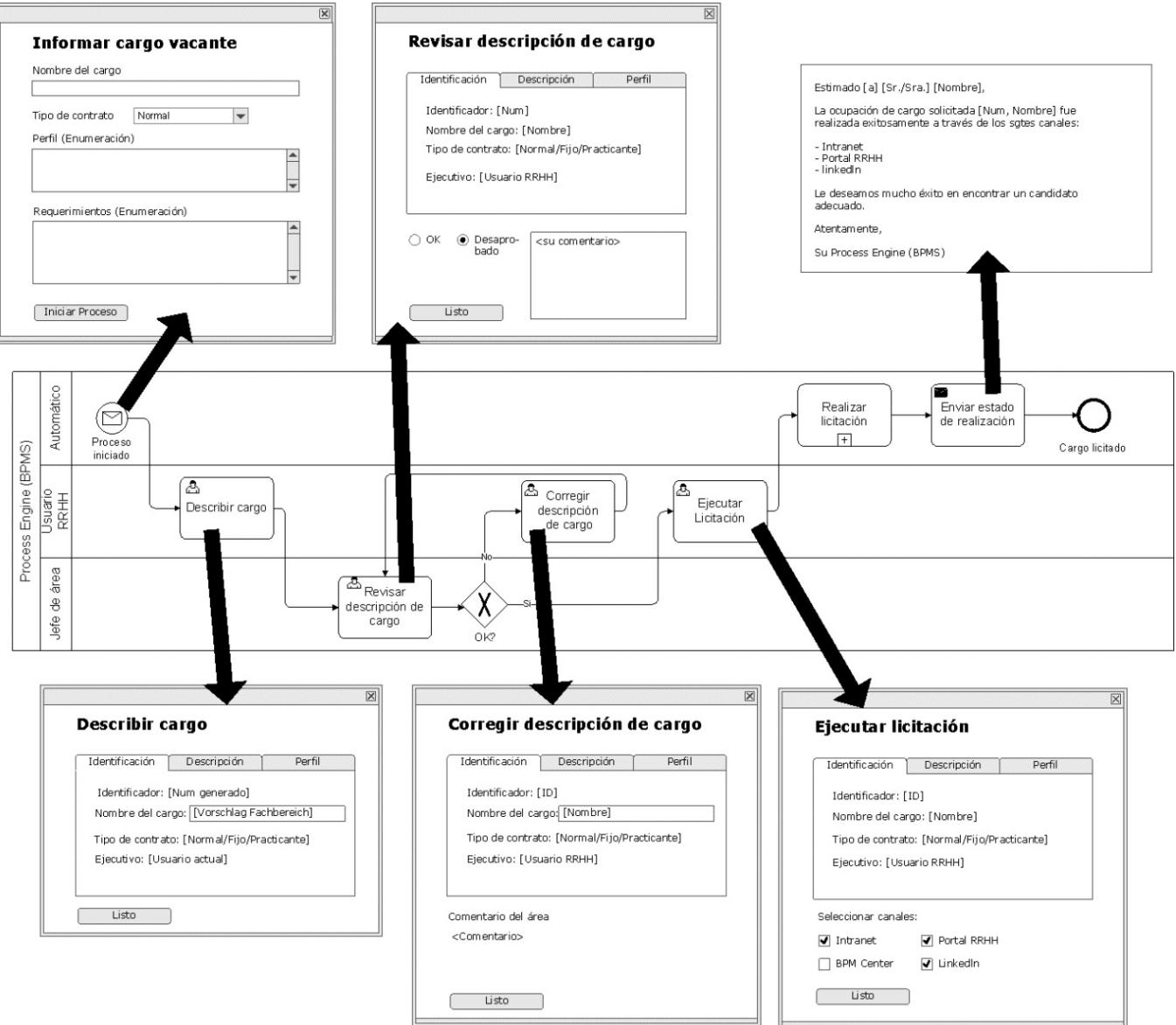


Figura 4.15: Prototipo de pantallas para el proceso de licitación de cargo
la interfaz de usuario
y la integración de datos.

Reglas de negocio: Reglas de negocio son algoritmos que por lo general son encapsulados en programas y que se pueden invocar a través de servicios. Sin embargo hoy en día existen motores de reglas, llamados en inglés Business Rules Management Systems (BRMS), que administran y ejecutan las reglas de negocio, independiente de las actividades en los procesos que las necesitan. Algunas Suite de BPM (BPMS) traen motores de reglas incorporados, en otros casos existen BRMS independientes que pueden ser invocados por sistemas de software. Cuales quiera que sean las opciones por las que su organización ha optado, lo importante es que no será necesario o mejor dicho no recomendamos de modelar las reglas en BPMN, sino sólo identificar donde se utilizan e invocar estas en las actividades que las requieran. BPMN 2.0 introdujo justamente para estos fines un tipo de actividad «regla de negocio», que indica en el diseño técnico el lugar donde hay que invocarlas. La sección 4.5.4 está dedicada a este tema.

Interfaz de usuario: Desde el punto de vista de la automatización de los procesos, la interfaz de usuario no representa otra cosa que el punto de conexión entre los usuarios y la lógica de los procesos en la ejecución de la secuencia de las actividades. Pero también la interfaz de usuario puede tener su lógica propia de secuencia que no es igual a la lógica del proceso. Desde el punto de vista metodológico, podríamos decir que también la lógica de la interfaz de usuario, es decir el flujo de pantallas, obedece a una lógica propia, que debería modelarse en un pool separado. Estrictamente hablando también el flujo de pantalla es un proceso. En una arquitectura SOA que implementa BPM, la capa de representación está separada de la capa de orquestación de los procesos y su grado de acoplamiento es bajo con respecto al motor de workflow. En BPMN tendríamos que modelar dos pools que colaboran a través de flujos de mensajes, uno para la interfaz de usuario y otro para el motor de workflow.

Algunos BPMS traen generadores de formularios de pantallas incorporados que se encuentran amarrados a las actividades del flujo, naturalmente este proceder es más rápido en el desarrollo del sistema, pero tiene la gran desventaja que es menos flexible al cambio, sólo la separación de estas capas y la integración a través de servicios logra la flexibilidad para reaccionar al cambio que se espera en BPM.

Integración de datos: Procesos intensivos en la integración de datos, también requieren de una arquitectura que cumpla con el principio de un bajo acoplamiento con los sistemas que suministran esta información. Aquí entra en juego la componente llamada «Enterprise Service Bus (ESB)», que nuevamente desde el punto de vista BPMN representa un pool propio, el cual orquesta los diferentes servicios con el motor de workflow. Lógicamente la implementación también puede hacerse desde las actividades directamente con los sistemas, pero con la desventaja de pérdida de flexibilidad ante cambios de cualquier índole, tanto técnicos como funcionales.

4.4.5. Implementación del flujo a través de desarrollo propio

El nivel 2 es independiente de la implementación técnica del nivel 3. Como hemos visto hasta el momento, en el nivel 3 (técnico) nos encontramos con muchas alternativas de implementación. La primera gran decisión es de apoyarse por alguna Suite de BPM (BPMS), sistema de workflow (WfMS) u algún otro Process Engine, como por ejemplo open source (figura 4.16, Nivel 3a), u optar por un desarrollo propio (Nivel 3b).

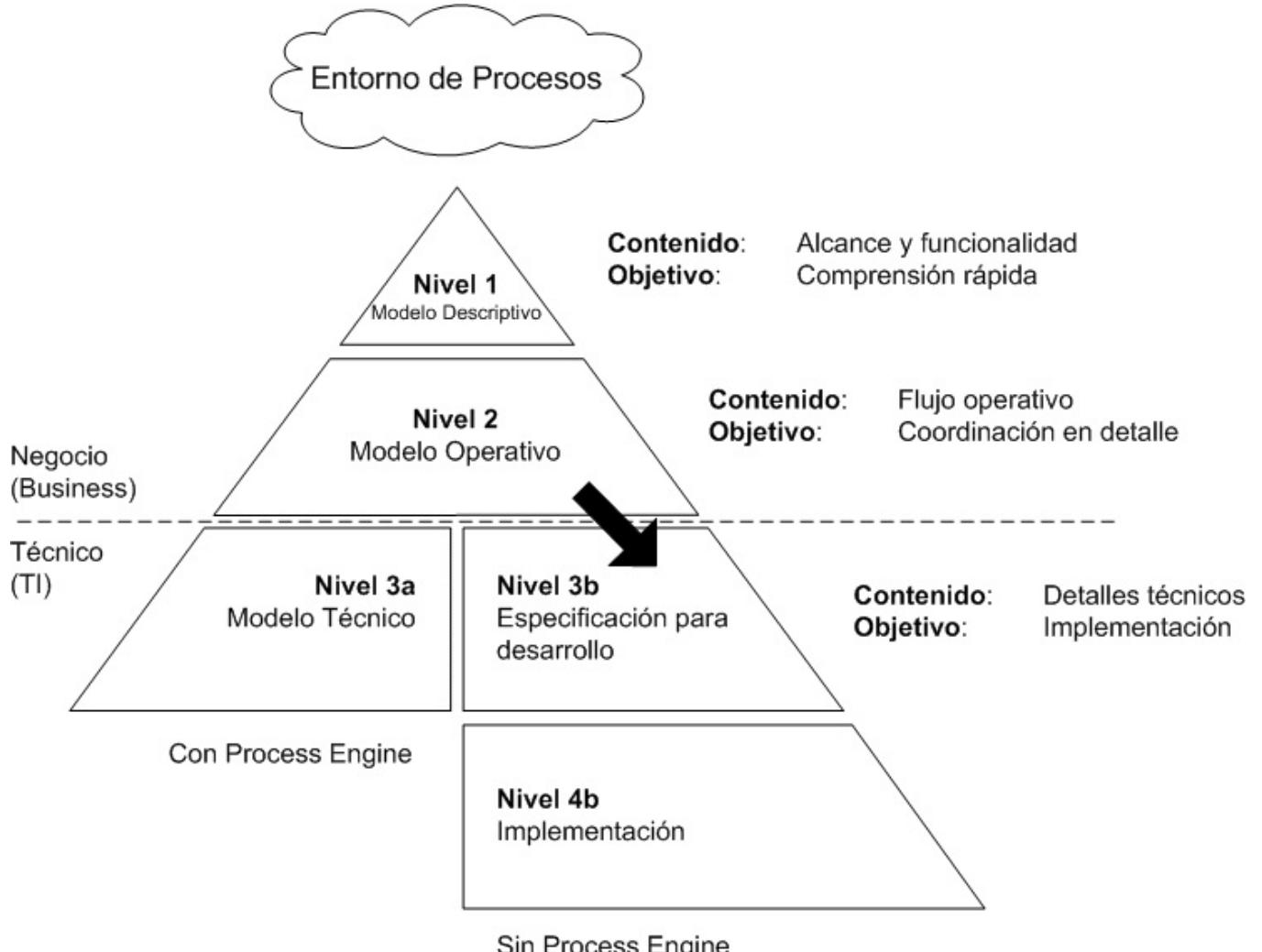


Figura 4.16: Transición del nivel 2 operativo al nivel 3b técnico

La más arcaica y antigua posibilidad es de programar toda la solución de workflow, en algún lenguaje como Java, .net o C# u otro (Nivel 3b). Entonces su «Process Engine» va a ser su compilador o su intérprete. A diferencia de la alternativa de utilizar una plataforma BPMS, en el caso de desarrollo propio no podrá traspasar directamente los procesos diseñados en el nivel 2 al 3. En este caso tendrá que al igual que en cualquier otro proyecto de desarrollo de software hacer primero una especificación para el sistema. Entonces el modelo de procesos lo tendrá que descomponer en casos de uso, y llevarlos a una especificación como se acostumbra representar con diagramas de actividad UML. A diferencia de los sistemas de workflow, las aplicaciones de TI representan una colección de funcionalidades que, dependiendo del proceso, el usuario las va ejecutando. En los sistemas de workflow, el sistema guía el proceso; en las aplicaciones es el usuario el que guía el proceso. Podríamos decir que las aplicaciones pueden encapsular subprocessos, pero no un proceso end to end, de lo contrario se habría convertido en un sistema de workflow.

Podemos resumir, que en un principio también se podría utilizar BPMN para desarrollar aplicaciones, pero esa no fue la finalidad de desarrollar la notación. También podemos resumir, que sería una aberración implementar una solución de

BPM con desarrollo propio.

4.5. Recomendaciones Prácticas para el Nivel 2 4.5.1. Del happy path a la cruda realidad

La técnica del First Pass Yield y su relación con BPMN Entre las técnicas de optimización de procesos se conoce la de «First Pass Yield (FPY)», podríamos traducirlo como «Rendimiento de primera pasada». Se entiende como el porcentaje de resultados correctos, en la primera pasada de un flujo de proceso, que no requieren de mayor intervención[Fis09].

Lo que se busca con esta técnica en la optimización de procesos es de maximizar el FPY, para el cual existen bastantes técnicas de análisis, que se vienen empleando hace ya muchos años. Sin embargo el problema principal de estos métodos es que están basados en indicadores, por ejemplo en porcentajes de error o ciclos de tiempos, cuyos datos deben ser levantados en forma manual o presumidos. Ambas técnicas de obtención de datos están sujetas a errores y requieren de un gran esfuerzo para levantarlas. Entonces sería un gran apoyo si pudiéramos integrar estos indicadores en BPMN, cuyos datos reales pueden ser suministrados por el sistema de workflow implementado.

Pero antes de integrar estas técnicas convencionales de optimización hay que entender su concepto tradicional, el cual vamos a explicar en el ejemplo de esta técnica. Veamos entonces el típico diagrama de flujo (ver figura 4.17) que se utiliza en esta técnica. Este pequeño proceso tiene un «flujo principal», que se recorre en forma secuencial de arriba a abajo hasta la entrega del resultado final. Si el resultado debe ser corregido, entonces el proceso debe tomar el flujo alternativo. Como el flujo principal es el normal, es decir el flujo deseado por la organización, se le llama también «Happy Path». La probabilidad que el resultado no esté ok y deba ser corregido es del 30 % en este modelo. Por otro lado en este negocio un 70 % de los casos siguen el «Happy Path» o el flujo principal. Este último es el FPY. Con estos datos podemos ahora deducir algunos indicadores. En este simple ejemplo obtenemos entonces:

First Pass Yield: 80 Minutos

Worst Case (peor caso): 125 Minutos

Promedio: 93,5 Minutos

El promedio se da: $FPY * 0,7 + Worst\ case * 0,3 = 56 + 33 = 93,5$ Minutos. A este tipo de cálculo se le llama cálculo de variantes, que para mantener el ejemplo simple no aplicamos procedimientos iterativos.

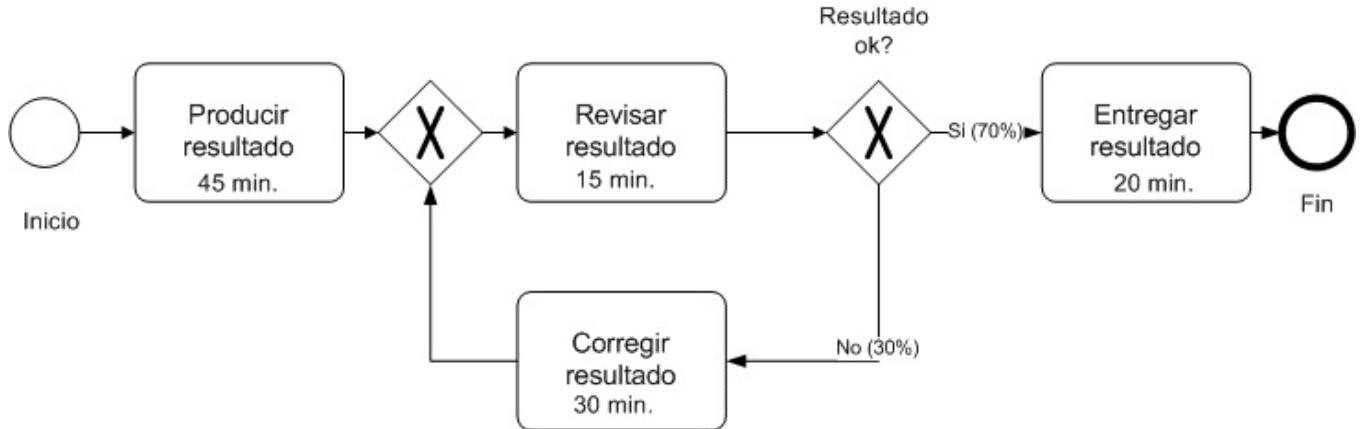


Figura 4.17: Diagrama de flujo simple par FPY en BPMN

¿Podríamos aplicar esta técnica de optimización también a BPMN? Como lo mostramos en la sección 2.13.5 se puede aplicar si agregamos los atributos necesarios a los elementos de BPMN y la herramienta tiene las funcionalidades correspondientes. En nuestro ejemplo Licitación de Cargo existen dos flujos del tipo bucle que en algunos casos alargan el tiempo de ciclo:

1. La descripción de cargo que se le entrega a Cristian no está completa y tiene que pedirle a Falko mayor información. Vamos a suponer que esta iteración sólo se hace necesario una vez.
2. La descripción de cargo aun no le parece a Falko y le pide a Cristian que la corrija.

En el diagrama de colaboración (ver figura 4.13) de nuestro ejemplo, queda claro que partes del proceso son administradas por el sistema, y cuales no. Por consiguiente sólo podríamos medir en forma automática aquellas partes del proceso que fueron implementadas:

1. El sistema puede medir los tiempos de ciclo de las actividades: «Describir cargo», «Revisar descripción de cargo», «Corregir descripción de cargo», «Ejecutar Licitación» y el subproceso «Realizar licitación».
2. El sistema también puede medir, cuantas veces fue necesario corregir la descripción de cargo.

Estos indicadores, los puede configurar en el panel de control del sistema o exportarlos a una tabla de Excel, agregarle algunas funciones adicionales como cálculos de promedio y generar gráficos que harán feliz al área de Control de Gestión o a sus jefes.

Pero en el modelo también vemos que el sistema no puede captar y medir todos los pasos del proceso de negocio. El sistema no se entera cuando Cristian llama a Falko para pedirle información, es decir no puede registrar estos pasos manuales que

quedaron fuera del control del sistema del proceso. Este hecho puede llevar a mal interpretaciones en la estadística que genera el sistema, por lo que tenemos tres posibilidades para tratar este problema:

1. Aceptamos y vivimos con esta inconsistencia, pero al menos sabemos donde se produce. En nuestro ejemplo en la actividad «Describir cargo».
2. Usted hace un levantamiento manual de los tiempos faltantes y la cantidad de iteraciones que se dieron y completa sus tabla estadística para evitar estas inconsistencias.
3. Se toma la decisión de incluir también aquellos pasos manuales, al sistema como «Human Workflow», entonces tenemos por lo menos todo el workflow bajo control y podemos hacer y registrar en forma automática todas las mediciones antes indicadas. Sin embargo esto no quita la posibilidad que tienen los usuarios, de seguir haciendo coordinaciones por fuera y no seguir los flujos alternativos que sugiere el workflow. Así por ejemplo Cristian y Falko podrían seguir coordinándose por teléfono, a pesar que se debería usar los pasos que provee el sistema.

Como usted se habrá dado cuenta la automatización de los procesos, puede convertirse en un instrumento muy poderoso y efectivo para fines de control de gestión, pero se recomienda no exagerar en automatizar el control minucioso y en detalle de los pasos de los usuarios, porque se van a sentir amarrados, observados y por último si los usuarios no siguen el flujo indicado, el sistema tampoco arrojará cifras fidedignas.

Modelamiento explícito de Errores: Como tratamos en la sección 2.6.4, tenemos en BPMN la posibilidad de modelar en forma explícita errores por medio de este tipo de eventos que nos disponibiliza la notación. La pregunta que nos planteamos ahora es cuando usarlos y cuando no. En la última sección tratamos en nuestro ejemplo los casos de volver a repetir las actividades, cuando falta información o cuando hay errores en la descripción de cargo. En estos casos no sugerimos modelar con los eventos del tipo error, porque estos eventos están pensados para el caso cuando el proceso realmente «falla», es decir el proceso no se puede llevar a término en forma normal. En nuestro caso, la actividad se repite, el tiempo de ciclo es más largo, pero no podemos concluir que debido a la falta de información momentánea no vamos a licitar el cargo. Sin embargo en algunos casos, no siempre podemos diferenciar claramente el tipo de errores. Sin duda alguna, existen algunas zonas grises.

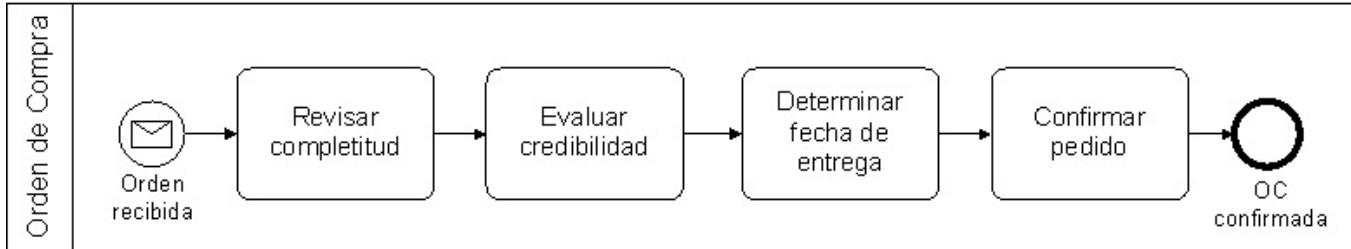


Figura 4.18: El proceso Orden de Compra en el «Happy Path»

Para tratar este problema vamos a analizar un ejemplo simple de un proceso de orden de compra. Nuestro proceso de Orden de Compra (OC) se descompone en cuatro pasos: Primero se revisa la OC en cuanto a correctitud y completitud. Luego se evalúa la credibilidad del cliente. Luego se averigua y determina una posible fecha de entrega. Finalmente se confirma la OC al cliente por fax. En la figura 4.18 vemos el «Happy Path» de este proceso. La pregunta que nos hacemos ahora es todos los errores que pueden suceder, cómo reaccionamos ante estos errores y cómo lo representamos en nuestro modelo de procesos. Para nuestros efectos de análisis, vamos a avanzar de atrás, desde el resultado, hacia adelante. El resultado en el «Happy Path» es, que confirmamos la OC. La pregunta que formulamos ahora es: ¿Y qué es todo lo que puede suceder para que no resulte la confirmación de la OC? Naturalmente pueden pasar muchas cosas, hasta durante la ejecución del proceso, terremotos, cortes de luz, etc.. Entonces tenemos que restringir el tratamiento de errores a eventos predecibles, que lógicamente podríamos hacer una selección demasiado amplia o muy restringida. En nuestro ejemplo nosotros nos decidimos a tratar los siguientes:

1. Los datos de OC están incompletos.
2. La OC no es legible.
3. El número de cliente en la OC está mal.
4. El cliente no tiene credibilidad.
5. El producto pedido no está disponible.
6. Al enviar el fax contesta una persona que enojada cuelga y cancela la línea telefónica.

Si tuvieramos que modelar todas estas eventualidades en nuestro proceso, ¿cómo lo haríamos? BPMN nos entrega dos alternativas para hacerlo: Si una actividad nos ha suministrado una información que no está ok, entonces lo modelamos con un Gateway del tipo XOR-Split a la salida de la actividad. Si la actividad misma no pudo ejecutarse, entonces necesitamos un evento del tipo error (ver figura 4.19). De acuerdo a esta metodología podemos modelar ahora los seis casos mencionados en la enumeración anterior.

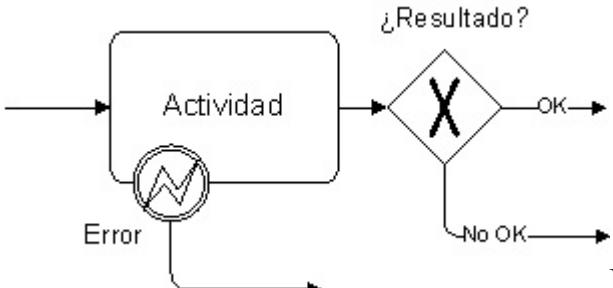


Figura 4.19: Alternativas para representar

problemas con BPMN

¡Antes de seguir leyendo haga usted el ejercicio de modelar estos casos y luego comparelo con nuestra solución!

En la figura 4.20 encontrará la propuesta de nuestra solución:

1. Los datos de OC están incompletos: Este caso es relativamente sencillo. La finalidad de la actividad es justamente revisar si los datos están completos, es decir la actividad fue ejecutada sin problemas. Entonces modelamos un XOR-Split a la salida de la actividad.
2. La OC no es legible: ¿Se puede revisar la OC si alguno de los datos o algunos de ellos no son legibles? No es tan claro como el primer caso, pero nosotros nos decidimos que «si». Si no podemos identificar una información, entonces pasa a ser lo mismo como que no hubiera información, por lo tanto podemos evaluarlo como datos incompletos. Lo que si se hace necesario de notificar al cliente porque no pudo ser procesada la OC.
3. El número de cliente en la OC está mal: Si la organización posee un sistema de identificación de cliente por medio de un número identificador, entonces si tendríamos un error porque la OC no se podría seguir procesando. Supongamos que el sistema extrae el número de cliente del CRM y con este mismo número se invoca un servicio para evaluar la credibilidad. Es posible que el CRM devuelva una variable que el cliente no existe, o lo que es peor que devuelva datos que pertenecen a otro cliente. Este caso lo modelamos como un evento de error, sobrepuerto a la actividad «Evaluar credibilidad».
4. El cliente no tiene credibilidad: La actividad se ejecutó, pero el cliente no tiene credibilidad, entonces modelamos claramente este caso con un XOR-Split a la salida de la actividad.
5. El producto pedido no está disponible: Este caso no es tan sencillo de diferenciar como los anteriores. Si queremos determinar la fecha de entrega, entonces primero tenemos que conocer la disponibilidad del producto, pero si no existe disponibilidad, cualesquiera sean las razones, no podemos determinar la fecha. Esto significa que la actividad no pudo ejecutarse, por lo tanto modelamos con un evento de error.

6. Al enviar el fax contesta una persona que enojada cuelga y cancela la línea telefónica: Seguramente el lector ya habrá aprendido, por lo que podrá responder a este caso.

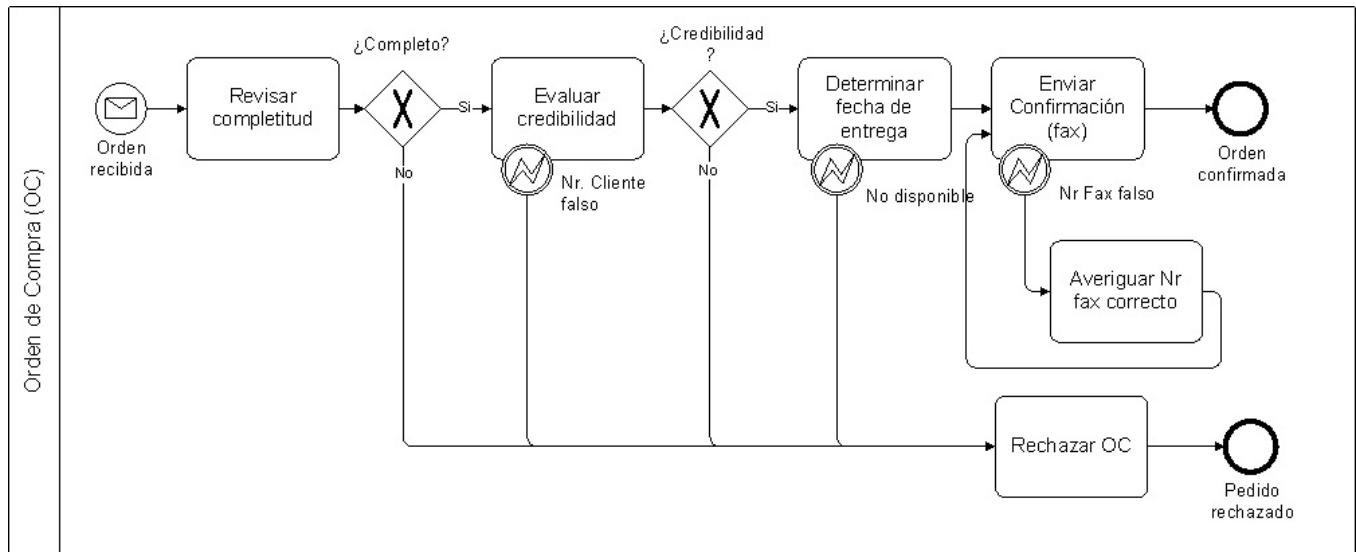


Figura 4.20: Alternativas de representación de errores desviadas del «Happy Path»

El lector atento se preguntará ¿por qué BPMN hace esta diferenciación?, y los errores no se tratan como en otras notaciones como fallas de cualquier índole y ¿se modelan todas con un Gateway del tipo XOR-Split? En un principio puede hacerlo, BPMN no se lo impide, pero nosotros le recomendamos hacer esta diferenciación por los siguientes motivos:

Cuando se levantan y documentan los procesos con los usuarios de negocio, generalmente están pensando en lo que sucede en operaciones, en el día a día, pero no en todo lo que puede suceder, sobre todo si pensamos en que hay que implementar una solución técnica. Entonces se modelan las típicas faltas que son propias del negocio, con XOR-Gateways. En el momento de pasar del nivel 2 al 3, es decir al diseño técnico, hay que repasar todas las eventualidades para evitar que el sistema se caiga o produzca inconsistencias. Generalmente se trata de sucesos técnicos o eventos impredecibles. Este tipo de errores los modelamos con eventos del tipo error, los diferenciamos claramente de los operativos. De esta forma podemos revisar nuevamente los modelos con los usuarios de negocio e identificamos inmediatamente los puntos por aclarar. En muchas ocasiones se rediseña el modelo, en el sentido que se pueden anteponer actividades de revisiones que evitan que se produzca un error técnico y no se pueda ejecutar una actividad. En nuestro ejemplo podría ser anteponer a la actividad «Determinar fecha de entrega», una actividad «Revisar disponibilidad de producto» y si no está disponible se informa al cliente sobre la situación. Finalmente quedan sólo errores realmente técnicos, que evitan la caída del sistema.

Como se ha dicho anteriormente, los eventos del tipo error, si suceden aseguran la

marcha del sistema a través de un tratamiento excepcional. BPMN ofrece por esta razón asegurar el sistema en diferentes niveles, a nivel de proceso, de subprocesso y a nivel de actividad.

Los XOR-Gateways fueron concebidos para diferenciar casos, pero no sólo para diferenciar casos normales de anormales, sino también para separar casos positivos. «¿El producto lo quiere en el color rojo o azul?» podría ser una pregunta en el «Happy Path» por ejemplo. Ahora ni sintácticamente ni visualmente podemos diferenciar en el modelo los XOR positivos o negativos. Los objetos del tipo eventos de error son unívocos, se identifican y reconocen inmediatamente y el sistema también los puede diferenciar de un XOR-Gateway.

Finalmente y resumiendo este párrafo, por las razones anteriormente expuestas, le podemos recomendar hacer uso de este instrumento que es mucho más expresivo que sólo utilizar XOR-Gateways.

4.5.2. El verdadero beneficio de los subprocessos

Nuestro framework de BPMN está formado de varios niveles y en cada uno de estos niveles tratamos los modelos de los procesos con diferente profundidad. En el nivel operativo, además de trabajar con diferentes niveles de agregación, sepáramos los participantes y los representamos en vistas propias. La pregunta que vamos a tratar ahora es: ¿En qué aspectos de la conformación de niveles y vistas nos puede apoyar el tipo de objeto subprocesso? En la sección 2.8 describimos que los subprocessos cumplen fundamentalmente con tres funcionalidades:

Encapsular lógica compleja en un mismo plano de un modelo de procesos.
Modularizar lógica que pueda ser reutilizable en otros procesos.

Definir un dominio en un proceso (en inglés: Scope) que pueda reaccionar como un todo ante un evento (eventos sobrepuertos).

Estas funcionalidades las puede utilizar en cualquiera de los tres niveles de nuestro framework BPMN. En nuestro ejemplo de la licitación de cargo formamos en el pool del sistema de workflow el subprocesso «Realizar licitación» con el objetivo de no tener que mostrar todas las interfaces que requiere esta lógica (ver figura 4.13). Ahora tendríamos la posibilidad de definir un tratamiento de error para aquellos casos que las interfaces no respondan en forma exitosa. Como seguramente no vamos a reutilizar este subprocesso en otros modelos, no deberíamos declararlo como subprocesso global. Por consiguiente este subprocesso está subordinado al proceso que lo contiene, es decir le «pertenece». Dependiendo de la herramienta que utilice, este proceso se podría cerrar y abrir en un mismo plano.

Como podrá apreciar el lector, no sólo utilizamos subprocessos para identificar un

área que contiene lógica independiente, sino para encapsular lógica en un mismo plano y de esta forma reducir la complejidad y poder diseñar los modelos en forma más compacta, razón por la cual encontraremos en un diagrama de cualquier nivel subprocesos de varios tipos. Si el lector no ha trabajado con ninguna otra notación, le parecerá obvio de aprovechar estas ventajas en el modelamiento, pero la mayoría de las otras notaciones sólo conocen la conformación de subprocesos de áreas funcionales.

En BPMN podemos combinar todas estas posibilidades en los tres niveles, desde la representación de mapas de procesos en el nivel descriptivo, usar diferentes grados de abstracción en el nivel operativo hasta el nivel técnico. De esta forma logramos también pasar de un nivel a otro en forma más comprensible.

4.5.3. Los límites de la formalidad

BPMN se basa en el supuesto que podemos representar la lógica de un proceso a través del flujo de control, completo y en su máximo detalle. Mientras más detallado sea el flujo, más reducimos los grados de libertad a los usuarios de negocio. Este principio lo podríamos llevar a tal extremo, que el flujo de control se parecería a la organización robotizada de una línea de producción en la industria. La tendencia de este enfoque no les gusta a algunos ejecutivos y ya se han formado algunos foros en internet que se encuentran discutiendo sobre «el sentido que pueda tener BPM bajo estas circunstancias». Casi podríamos hablar de una discusión ideológica, en que se ha formado un bando en contra de la automatización de los procesos, el que podríamos llamar los «Humanistas» que se quieren resistir a la tendencia «tecnocratizar» su puesto de trabajo.

En este libro no queremos llevar a cabo esta discusión, pero si queremos comentar estos límites de la formalidad. En muchos proyectos de BPM nos encontraremos con zonas grises, situaciones que no se encuentran definidas o estandarizada la forma de proceder. Principalmente nos encontraremos con dos situaciones:

No existe claridad de cómo se ejecutan las tareas y la forma de trabajar no está organizada. Por alguna otra razón el conocimiento no está disponible. ¡Este estado lo queremos cambiar!

Existe el conocimiento experto por parte de los usuarios claves y sólo ellos saben como aplicarlo, pero no pueden o no quieren documentarlo. Por lo general aceptamos esta situación.

En el primer caso tenemos que documentar primero un estado intermedio indeseado, que lo podemos hacer por medio de la definición de un subproceso. En el segundo caso documentamos los productos o servicios que entregan los usuarios claves. Para

poder modelar ambos casos podemos apoyarnos de un tipo especial de subprocesso, llamado «Ad-hoc», el cual definimos en la sección 2.8.4.(ver también figura 4.21).

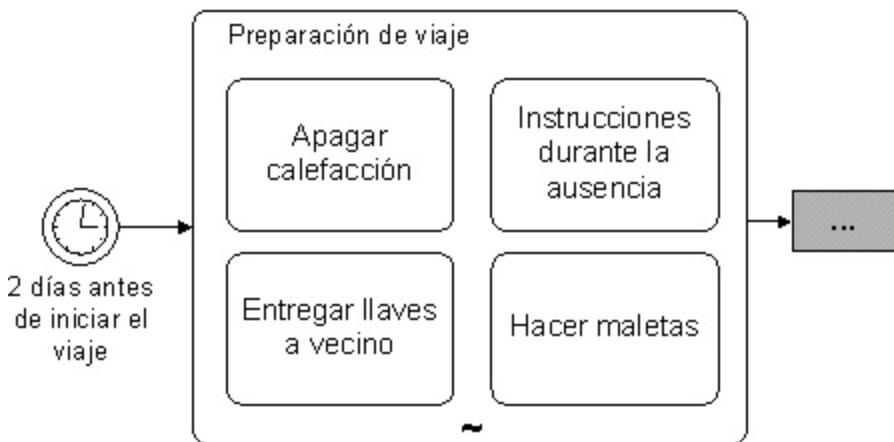


Figura 4.21: Caso con uso

de subprocesso con propiedad Ad-Hoc

Este tipo de subprocesso es como una carta en blanco para el analista de proceso: Se enumeran los actividades o subprocessos que pueden ejecutarse, pero el subprocesso no especifica la lógica entre ellos. Es decir, no especifica nada respecto que se hace primero o después y cuantas veces se pueden repetir algunas o todas las actividades o subprocessos contenidos en el subprocesso Ad-hoc. La decisión de toda esta secuencia la toma el responsable del subprocesso. Estos subprocessos del tipo Ad-hoc, también los podemos utilizar cuando una situación es poco clara en el levantamiento de una situación actual. En este caso enumeramos y relacionamos sólo lo que sabemos y dejamos este subprocesso como un paquete de trabajo pendiente que debe ser modelado. Pero también lo podemos dejar para el segundo caso, documentar las tareas de los expertos. Ellos sabrán cómo y cuando las ejecutan.

Si usted piensa implementar estos subprocessos Ad-hoc, podríamos tener problemas porque nos estamos saliendo de los límites del BPM o bien de lo que se puede representar en sistemas de workflow. Más bien nos encontramos con disciplinas afines como «Case Management», que en tradicionales motores de workflow no se pueden representar. En cierta forma el subprocesso Ad-hoc, tiene un parecido a los clásicos diagramas de «Use Case» en UML, en el cual sólo se enumeran las funciones (la enumeración de casos en aplicaciones por lo general se diseña como un menú en las pantallas) que el usuario puede seleccionar en su aplicación de software, pero el sistema no le dice cómo y en qué orden las ejecuta. Entonces es muy posible que al automatizar procesos que contienen subprocessos Ad-hoc, tenga que prever «cajas negras» en reemplazo de la faltante lógica de negocio.

4.5.4. Independizar las reglas de negocio de los procesos

En la sección 4.5.1 tratamos en el ejemplo del proceso «Orden de compra» los tipos de errores que pueden ocurrir. Para este mismo caso vamos ahora a formular la

pregunta: ¿Bajo qué condiciones habrá que revisar la credibilidad del cliente, antes de confirmar la OC? Vamos a suponer que la respuesta a esta pregunta va a depender del tipo de cliente y el volumen del pedido, expresado en dólares. Entonces podemos definir el primer paso en el proceso como una actividad de «Revisión de datos de la OC» y en esta actividad se va a decidir, si es necesario hacer una evaluación de credibilidad o no (ver figura 4.22).

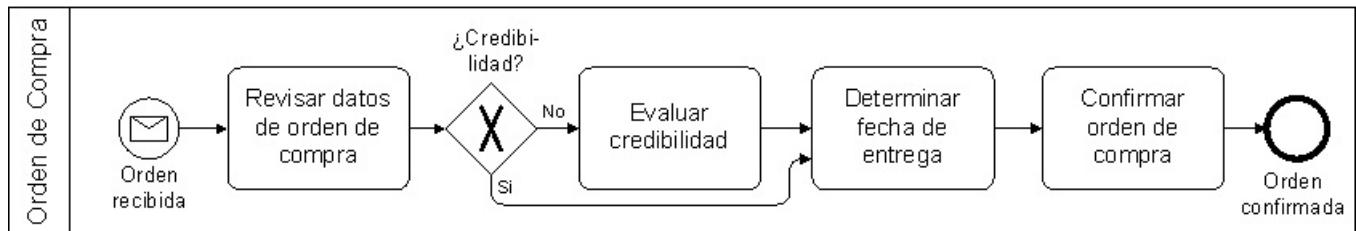


Figura 4.22: Proceso Orden de Compra con revisión de credibilidad

En el siguiente paso tenemos que conocer las condiciones concretas para representar las regla de negocio:

Habrá que revisar la credibilidad del cliente, si el valor del pedido es mayor a \$ 300.000.

Si se trata de un cliente nuevo, habrá que revisar la credibilidad del cliente a partir de los \$ 50.000.

Si se trata de un cliente VIP, no es necesario revisar la credibilidad del cliente. Con esta información podemos modelar la regla de negocio como lo muestra la figura 4.23.

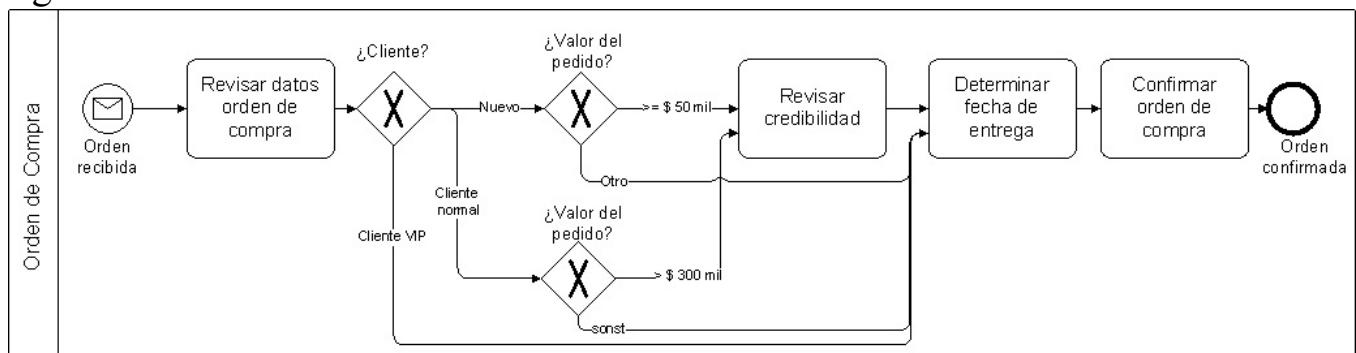


Figura 4.23: Proceso Orden de Compra con regla de negocio modelada

¿Qué le parece al lector este diagrama? Imagínese ahora que cambian las políticas de negocio y haya que incluir otras condiciones adicionales. Seguramente usted estará ya pensando en las consecuencias que tendría la ampliación para el modelo:

Cada nueva condición implica un nuevo Gateway y flujos de secuencia.

El problema aumenta, si las condiciones están entrelazadas, como lo es nuestro caso de tipo de cliente y valor del pedido.

El diagrama del proceso se vuelve rápidamente ilegible.

Si cambian las reglas frecuentemente habrá que estar adaptando el modelo a la nueva

situación y lo mismo sucede cuando las reglas están embebidas en los programas computacionales..

Si además tenemos que revisar la credibilidad del cliente en otros procesos, por ejemplo para entregar una cotización, habrá que modelar y administrar estas condiciones en forma redundante.

Podemos concluir entonces, que modelar condiciones que representan «reglas de negocio» no es una «buena práctica» en el modelamiento de procesos. Para evitar que esto suceda, el analista debe aprender a diferenciar entre «reglas de negocio» y «reglas de ruteo (en inglés routing: guiar, enrutar)».

Gestión por Reglas de Negocio (en inglés: Business Rules Management) es una disciplina separada y como el nombre lo indica reglamenta las condiciones del negocio. En Gestión por Procesos de Negocio, la administración centralizada e independiente de las reglas de negocio toma la importancia de un factor crítico de éxito para el negocio. Tanta es la importancia que se les da en algunos rubros a las reglas de negocio, que hace un tiempo ya se han formado comunidades de reglas de negocio, que incluso publicaron un «Manifesto de Reglas de Negocio» [SG06] [BRG10][BRC10]. Hoy en día existen sistemas de software especializados e independientes que administran y ejecutan las reglas de negocio (en inglés: Business Rules Management System (BRMS)). Estas funcionalidades pueden ser invocadas desde un sistema de workflow o BPMS, pero también muchos BPMS traen editores de reglas, aunque este no sea independiente del sistema.

En el modelamiento de procesos tenemos que distinguir claramente entre reglas de negocio y reglas de ruteo, siendo sólo estas últimas las que se deben modelar. Para editar y mostrar reglas de negocio por lo general se usan tablas de decisión. ¿De qué forma se tratan entonces reglas de negocio en un modelo de procesos? Para estos efectos volvemos a nuestro ejemplo de la «Orden de Compra». Traspasamos las condiciones y las reglas de nuestro ejemplo a una tabla de decisión como el lector puede apreciar en la figura 4.24.

Condiciones		Decisión
Tipo de cliente	Valor	Revisar credibilidad
Cliente VIP	no aplica	No
	> 300.000 \$	Si
Cliente normal	<= 300.000 \$	No
	>= 50.000 \$	Si
Cliente nuevo	< 50.000 \$	No

Figura 4.24: Tabla de decisión

para reglas de negocio

¿Cómo relacionamos ahora la tabla de decisión con nuestro modelo del proceso? A modo de ejemplo muestra la figura 4.25 como deben modelarse reglas de negocio en

BPMN:

Antes del XOR-Gateway se introduce una actividad especializada que no tiene otra función que ejecutar las reglas de negocio asignadas.

El resultado de la actividad es la decisión, de lo que debe hacerse a continuación.

El XOR-Gateway ahora sólo tiene la función de ruteo a través del flujo de secuencia. La asociación a la tabla de decisión la podemos hacer en los atributos de la tabla, o podemos asociar un objeto de datos tipo input a la actividad.

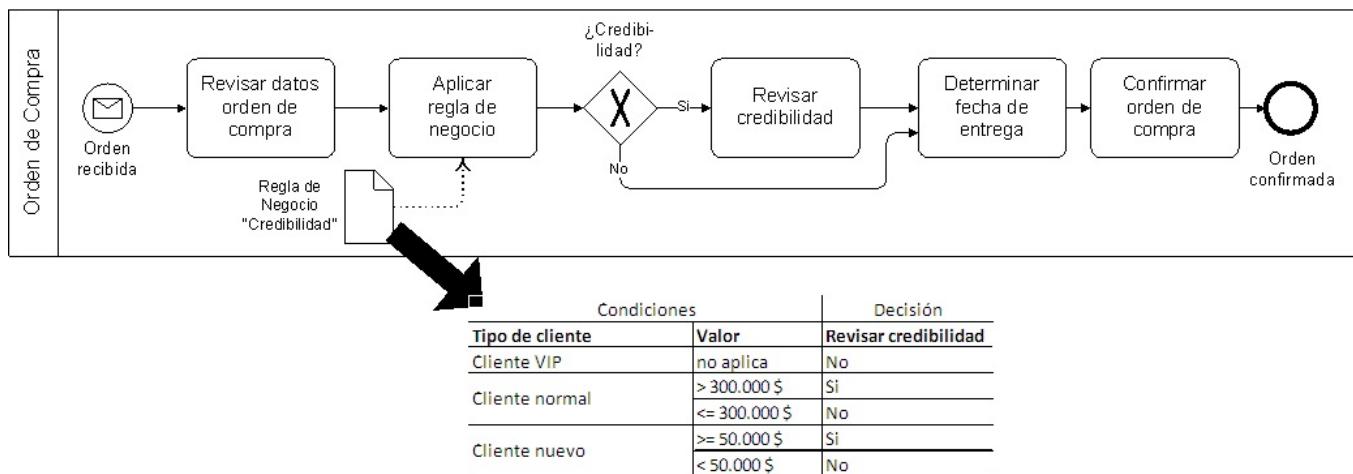


Figura 4.25: Forma de representar reglas de negocio en modelos de BPMN

Nuestro Tip en BPMN

La asociación mostrada con la tabla de decisión no es parte de la notación, sino que tiene que ser una funcionalidad que provee la herramienta. La especificación de BPMN permite objetos de datos específicos o declarar atributos con propiedades individuales. Los proveedores de herramientas aprovechan generalmente estas posibilidades y ofrecen funcionalidades como adjuntar documentos o linkear URL con archivos que se encuentran en algún servidor de la red, o bien URL que derivan a una página de intranet. Los administradores de reglas de negocio pueden entonces mantenerlas en forma centralizada y el usuario puede acceder a través de los modelos a la reglas actuales.

A continuación describimos las diferencias de ambos tipos de reglas:

Reglas de ruteo son evaluadas por XOR-Gateways, OR-Gateways o flujos de secuencia condicionales. Las reglas de ruteo son generalmente estables, simples y no entrelazadas.

Reglas de negocio pueden ser muy complejas y se administran en forma independiente del modelo de proceso. La regla de negocio puede entregar la variable que controla el flujo de la regla de ruteo.

Con la finalidad de diferenciar estos casos BPMN 2.0 ha definido una actividad del

tipo regla de negocio, como lo explicamos en la sección 2.7 y lo muestra la figura 4.26. En nuestro ejemplo sería nuestra actividad «Aplicar regla de negocio» justamente una actividad de este tipo que podríamos asignar a partir de la nueva versión. La OMG introdujo justamente este tipo de actividad para promover la separación de «modelos de procesos» y «modelos de reglas». Con las áreas de «Semantics of Business Vocabulary and Business Rules (SBVR)» define la OMG incluso un lenguaje propio para el modelamiento de reglas, pero este tema corresponde a un área especializada en el mundo de las reglas. En general la mayoría de las reglas de negocio se pueden representar sin mayores complicaciones por medio de tablas de decisión.

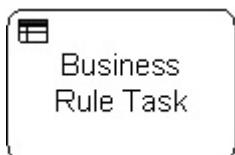


Figura 4.26: Actividad tipo regla de negocio

Importante de mencionar también en el contexto de la aplicación de reglas de negocio en modelos de proceso BPMN, es que podemos asociar un evento de condición a un sistema de reglas de negocio. Desgraciadamente BPMN aún es pobre en cuanto al grado de expresividad del evento de condición. Desde el punto de vista técnico podríamos interpretar la asociación de una regla a un evento de condición, ya que un motor de reglas se encuentra continuamente revisando si sucede el evento que cumple con la condición de la regla especificada. Si ocurre, el motor de reglas da el aviso al sistema de workflow sobre el evento acaecido de forma que el evento de condición asociado a la regla, gatilla el inicio o la continuación del proceso (ver figura 4.27).

Para los procesos de alineamiento entre las áreas de negocio y de TI es importante que entienda los principios detrás de esta estructura porque estos aspectos ¡no se limitan sólo a la implementación técnica! Dependiendo del caso de negocio el modelo también responde a una solución manual, como por ejemplo que una persona esté monitoreando el cumplimiento de normativas de calidad, tributarias, de seguridad, etc. y si suceden el proceso tiene que reaccionar.

La combinación de BPM y BRM tiene actualmente un gran potencial que puede explotar para incrementar el grado de agilidad en los procesos. Hoy en día se sabe que los flujos en los procesos son relativamente estables, no así las reglas de negocio que se encuentran constantemente sometidas al cambio. Con la aparición de los BRMS las áreas de negocio pueden administrar en línea los cambios de reglas sin impactar en los procesos. Debido a lo anterior, le recomendamos tomarse el tiempo necesario para conocer más en detalle esta disciplina[BRC10][BRG10].

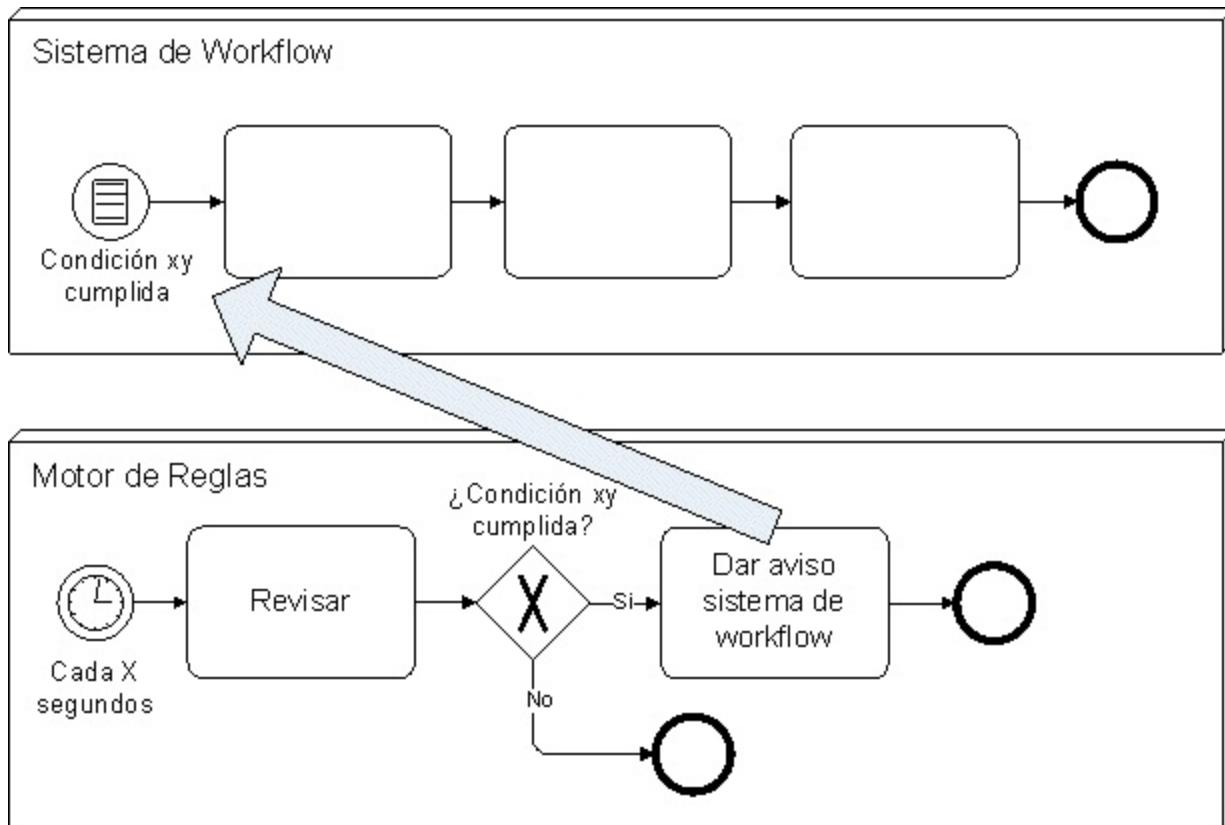


Figura 4.27: Evento de condición y Motor de Reglas

4.6. ¿Filtros en el uso de Símbolos?

Para la elaboración de los modelos en el nivel operativo no queremos hacerle ninguna recomendación en cuanto a restringir el uso de símbolos de la gran paleta que ofrece BPMN 2.0. El uso de los objetos en el nivel 2 depende mucho del tipo de negocio. Así por ejemplo puede ser extremadamente importante para un modelo de negocio, modelar una transacción con eventos de compensación. A pesar que para la mayoría de los procesos de negocio no será necesario emplearlos, en algunos casos incluso pueden llegar a tener la importancia de factores críticos. Debido a esta razón no podemos descartar ningún objeto en el nivel operacional, como lo pudimos hacer para el nivel descriptivo. Si usted conoce su negocio lo puede hacer para su organización, pero en un principio no podemos hacerlo en forma genérica.

En la figura 4.28 le presentamos una tabla, que le puede servir al lector como orientación, de los objetos que hemos utilizado con mayor o menor frecuencia hasta el momento en nuestros proyectos de modelamiento de procesos. También agregamos a esta tabla los nuevos objetos de la versión BPMN 2.0, pero como la nueva versión es muy reciente, no hemos tenido mayor experiencia en el uso con ellos. Esto puede cambiar en el futuro.

En el verano del año 2008 hicimos en Alemania una encuesta a 127 analistas de BPMN, sobre la frecuencia de uso de estos objetos. El resultado de la encuesta fue

bastante diferente a la que nosotros le presentamos, en el sentido que se usaban muchos menos objetos en el nivel operacional que los disponibles o los que nosotros empleábamos en nuestros proyectos. Al preguntar de por qué no se utilizaban, la mayoría respondió porque no los conocían y que utilizaban los mismos objetos comunes que tienen la mayoría de las notaciones de procesos. Al respecto tenemos que admitir que en 2008 casi no existía literatura sobre BPMN más que la especificación misma en inglés. Cursos de BPMN tampoco existían muchos y para que decir muy pocas herramientas habían implementado BPMN como la especificación lo indica, pero la situación cambió en menos de tres años rápidamente. Actualmente existe literatura sobre BPMN, mucha oferta de capacitación y los proveedores de tecnología están implementando la nueva notación.

	Casi nunca	Raras veces	De vez en cuando	Frecuentemente	Siempre
Swimlanes					
Subprocesos					
Actividades (marcas)					
Actividades (tipo)					
Flujo					
Gateways					
Eventos de inicio					
Eventos intermedios					
Evento de término					
Artefactos					
Nuevo en BPMN 2.0					

Figura 4.28: Frecuencia de uso de objetos BPMN en el nivel operacional

Capítulo 5

Nivel 3: Modelo de Procesos Técnico

5.1. Sobre este Modelo 5.1.1. Objetivos y Beneficios

El objetivo del tercer nivel es finalmente automatizar los procesos por medio de software. En el capítulo anterior explicamos que esto puede hacerse por medio de desarrollo de software. Sin embargo es mucho más efectivo y más simple hacerlo con un Process Engine (ver figura 5.1) . Como sucede esto último se lo vamos a presentar en este capítulo.

Modelos técnicos bien diseñados pueden ejecutarse directamente en el nivel 3 con un Process Engine. Dicho de otra forma, el modelo técnico (diagrama técnico) es sinónimo de código fuente para el Process Engine, lo que tiene una importante implicancia: Los modelos tienen que estar correctos y lo suficientemente detallados como para ejecutarse. Lo que en el modelo no está formalmente claro, no lo puede interpretar el Process Engine.

Tiene una gran ventaja que los modelos técnicos puedan ser interpretados como código fuente. Los modelos técnicos reflejan siempre la situación actual, por lo menos aquellos que han sido implementados. Si se requieren cambios habrá que adaptar el modelo de procesos. Justamente ahora llegamos a un segundo aspecto muy importante de nuestro Framework: Si los modelos del segundo y tercer nivel se encuentran bien integrados, tenemos la gran oportunidad de mantener los modelos

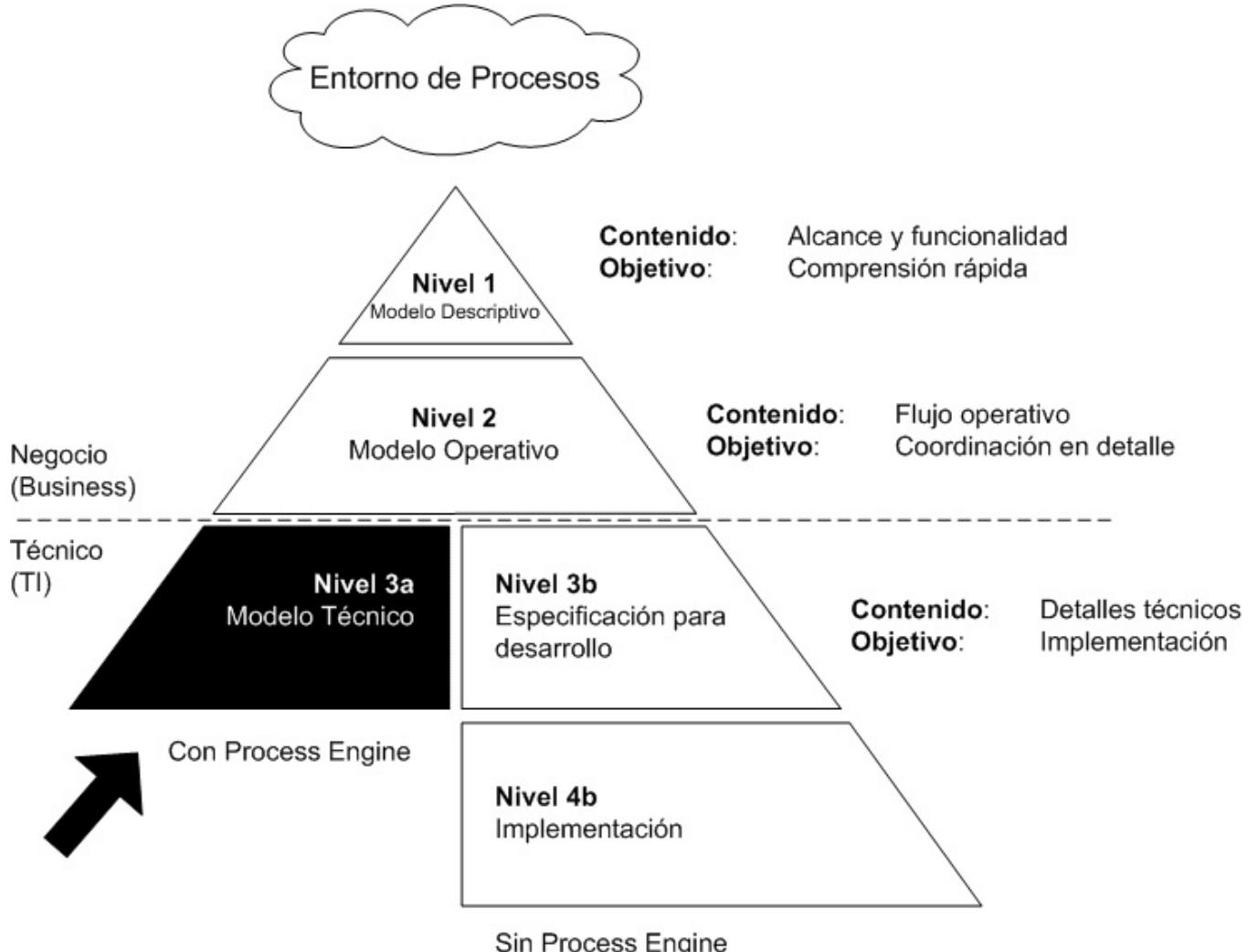


Figura 5.1: Nivel 3 Modelo Técnico del marco estructural para BPMN

de negocio en el nivel 2 actualizados. ¿Sabe el lector lo que esto significa? ¡Nada menos que cerrar la brecha entre la capa de negocio y la técnica, un sueño que hasta el momento no se ha cumplido!

Herramientas poderosas pueden exportar cambios desde el modelo técnico a los modelos de negocio y también datos que ha medido el Process Engine a los modelos de negocio. Si los modelos pueden importar estos datos en los atributos destinados para indicadores, se puede analizar el comportamiento de los modelos como realmente ocurren, en los mismos diagramas del modelo. Tenemos claro que existen pocas plataformas que estén tan bien integradas, como para que realmente funcione este ciclo, pero también es necesario que los modelos desde el punto de vista sintáctico y semántico se encuentren perfectamente alineados, un desafío metodológico pero realizable. Nosotros estamos convencidos que con la metodología mostrada en este libro, se puede lograr este alineamiento entre los modelos del nivel 2 (operacional) y el nivel 3 (técnico). ¿Esta es una de las brechas de la que tanto se 5.1. SOBRE ESTE MODELO 205

habla entre la capa de negocio y la capa de TI? ¡Nosotros creemos que si podemos cerrarla!

5.1.2. Requerimientos para el Modelo

En el nivel 3 los diagramas de procesos no sólo deben estar sintácticamente y semánticamente correctos, sino también tienen que considerarse todo el resto de aspectos técnicos que son necesarios para automatizar procesos con un Process Engine, por ejemplo casos de excepción y errores técnicos; mal que mal estamos hablando de código fuente.

5.1.3. Forma de proceder en el nivel 3

La forma de proceder en el nivel tres constituye un factor crítico de éxito para el proceso de automatización, porque justamente aquí impacta fuertemente el modelo de negocio en la implementación. Nuestra experiencia hasta el momento nos muestra que las posibilidades de éxito dependen fuertemente de las competencias y habilidades del analista de negocio y su capacidad de entenderse con el ingeniero de procesos (implementador).

El procedimiento incluye por lo general los siguientes pasos:

1. Aclarar y validar el modelo de proceso «to be» en el nivel 2. Este punto lo tratamos extensamente en el capítulo anterior.
2. Decisión sobre una plataforma y arquitectura tecnológica (Nota: Normalmente la definición y disposición de una plataforma tecnológica debería ser un proyecto separado de informática y ser un input dado para un proyecto de BPM).
3. Si se utiliza una Engine BPMN 2.0, sólo debería ampliarse el modelo de procesos en aspectos técnicos. Si se utiliza otro tipo de modelo técnico (BPEL, XPDL u otro) se hará necesario incluir una nueva capa y realizar un mapping entre el nivel 2 y el nivel 3.
4. Rediseño iterativo del modelo del nivel operativo en caso que surgan problemas que no permitan implementar el modelo de negocio.
5. Testear y ejecutar marcha blanca de acuerdo a los métodos tradicionales del ciclo de desarrollo de software.

Hasta el momento nos hemos referido sólo a los aspectos de la implementación de la lógica del modelo de procesos (flujo de secuencia). Aún no hemos hablado de otros requerimientos técnicos que son necesarios de cumplir para automatizar los procesos. En el modelo de negocio aún faltan por especificar:

Estructura de datos de acuerdo a la tecnología que se usará, XML, java, etc. Definición de la interfaz de servicios, por ejemplo webservices. Detalles del Human-Task, por ejemplo, roles, grupos de usuarios, permisos.

Antes de entrar en estos detalles técnicos queremos darle al lector una introducción general de cómo se entiende y se procede cuando se trata de automatizar procesos en base a modelos con un Process Engine.

5.1.4. Indicaciones para la lectura de este Capítulo

Consideramos necesario hacerle algunos comentarios al lector para la lectura de este capítulo que es eminentemente técnico. Debido a esta razón no traducimos términos técnicos del inglés al español; hacerlo significaría crear confusión entre los técnicos, porque sencillamente estos términos no se traducen en la práctica. Otro punto que no podemos cambiar o hacer más legible es en algunos ejemplos de mostrar el código fuente de modelos BPMN 2.0 en XML. Estos aspectos van a ser para muchos lectores incomprensibles o muy técnicos. Al lector que no le interesen estos detalles técnicos le recomendamos saltarse en parte o todo el capítulo. De todos formas tratamos de ocupar ejemplos sencillos y muy reducidos con el objetivo de no abultar las páginas de este libro. Por otro lado no quisimos desistir de mostrar en algunos casos el código XML, porque encontraremos también entre los lectores ingenieros o profesionales técnicos que se interesan en estos aspectos. Nosotros mismos al escribir este libro nos dimos cuenta que casi no existían ejemplos BPMN-2.0-XML desarrollados. Debido a lo anterior trabajamos en la OMG en un documento de ejemplo, que será agregado en forma oficial como anexo a la especificación. Entregamos a la OMG para estos fines un ejemplo completo e integrado, incluyendo el completo código fuente completo y hemos implementando este ejemplo en un proyecto Open Source BPM-Platform Activiti, una Process Engine que implementa directamente BPMN 2.0. El técnico interesado podrá ejecutar estos ejemplos en el sitio del proyecto y analizar el código fuente.

5.2. Fundamentos 5.2.1. Automatización de Procesos con un Process Engine

El «Business Process Engine» o en forma abreviada «Process Engine» es un componente de software que sirve para ejecutar procesos de negocio, también llamado «Process Execution» o automatización de procesos. El Engine necesita para ejecutar directamente los procesos un modelo que contenga todos los detalles técnicos que se requieren para ello. Una vez implementados los procesos, el Engine ejecuta para cada uno de los procesos las instancias que recorren el flujo de negocio. El Engine mantiene el control por separado de cada uno de ellos, ya que sabe lo que tiene que hacer primero y después de cada actividad que se recorre en la lógica modelada. El concepto de Token descrito en la sección 2.1.4 es realmente implementado y es el concepto técnico fundamental para mantener el control de cada una de las instancias.

El Process Engine distingue principalmente dos tipos de actividades: aquellas en que

la interacción con el ser humano es necesaria y todas las otras que se ejecutan en forma automática, por ejemplo invocación de servicios, o la evaluación de las variables de Gateways y eventos. Para la interacción con el ser humano se emplean actividades del tipo manual. Generalmente existe un listado de tareas para el usuario, similar a un administrador de correos entrantes de e-mail. Si el usuario selecciona y ejecuta una de estas tareas entrantes se abre una pantalla preconfigurada, en la cual se ven los campos con las datos que han de administrarse o sobre los cuales el usuario tendrá que tomar decisiones. En forma esquemática muestra la figura 5.2 como trabaja un Process Engine.

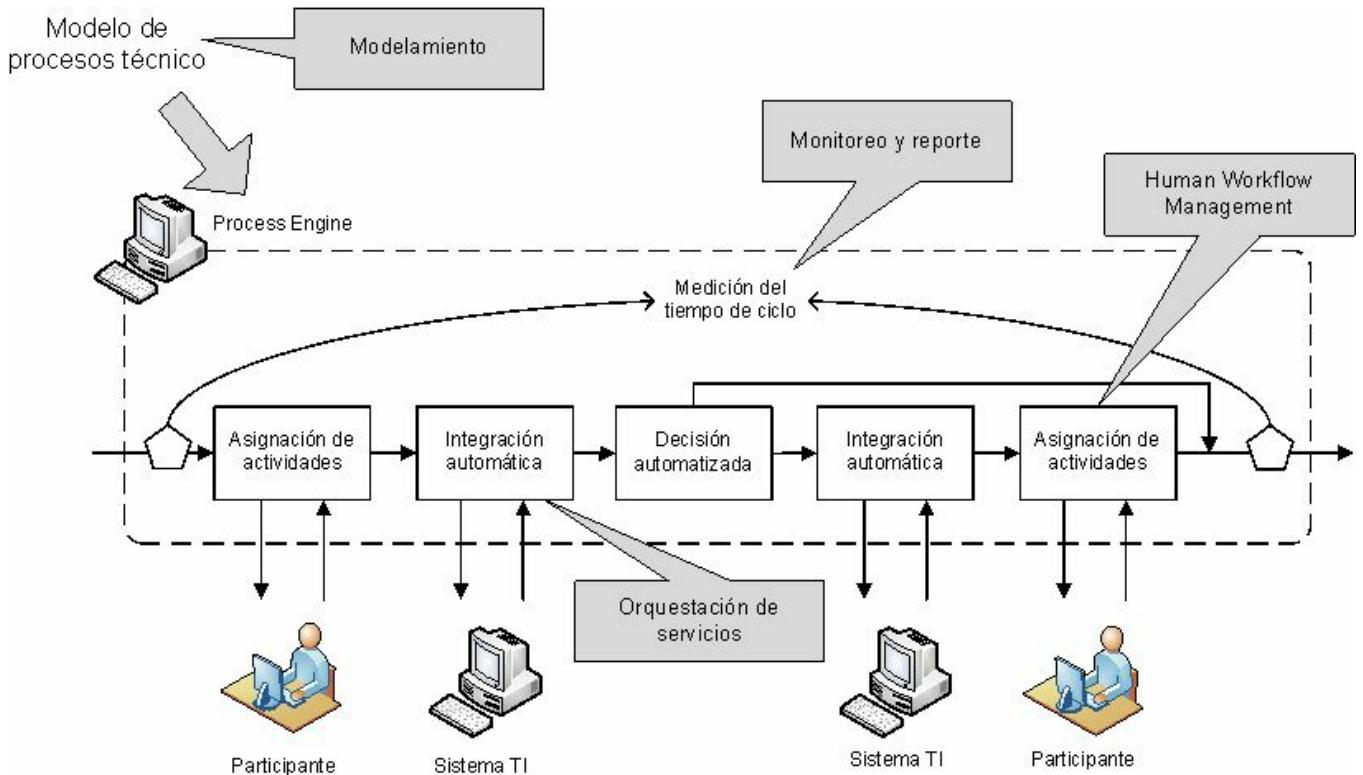


Figura 5.2: Forma de trabajo de un Process Engine

En el ejemplo de las actividades manuales, queda claro que el Engine tiene más funcionalidades que sólo automatizar el flujo de control, porque también tiene que administrar el flujo de datos propio del proceso. Así por ejemplo durante una instancia, se pueden crear datos que el Engine tiene que administrar junto a los estados de las actividades. Estos datos deben ser almacenados en forma persistente en alguna base de datos y asegurar su permanencia ante una caída del sistema.

A continuación enumeramos las típicas funcionalidades que debe proveer un Process Engine, más allá de administrar el flujo de control y el flujo de datos:

Versionamiento de modelos de procesos: por su naturaleza los procesos de negocio por lo general tienen un ciclo bastante largo. Un Proceso que administra una orden de compra puede durar entre varios días y hasta meses, dependiendo del tipo de negocio, pero esto significa también en cada instante en que se realiza un cambio, existen aún instancias activas. Debido a lo anterior, queda claro que un Process

Engine debe manejar bien el control de versiones de múltiples instancias y de modelos simultáneamente.

Registro de datos (logs), indicadores y evaluaciones: el Process Engine debe poder registrar todos los datos durante la ejecución de las diferentes instancias (en inglés: log files). Así se puede hacer un seguimiento de todo lo que ha sucedido en cada una de las actividades ejecutadas, por ejemplo cuando se aprobó la orden compra (fecha y hora), cuando se despachó el producto, etc.. Estos datos de log también se pueden agregar, filtrar y en general evaluarse para fines de optimización y control de gestión, por ejemplo para detectar cuellos de botella. Por lo general estos datos son puestos a disposición en una componente de gestión llamada «Business Activity Monitoring (BAM)», un cuadro de mando a nivel de usuario y supervisor que también permite configurar alertas, envíos de email y hasta reconocer patrones indebidos.

Administración y monitoreo técnico: El Process Engine permite visualizar el estado momentáneo de las instancias de los procesos activos. El administrador tiene la facultad de intervenir en estas instancias, por ejemplo de reiniciarlas, de cancelarlas, de transferirlas a otros usuarios etc.

Los modelos de procesos deben disponibilizarse para el Process Engine en un lenguaje adecuado. La pregunta más actual en el mercado de los Process Engines, y que vamos a tratar en las siguientes secciones, es si los modelos de negocios deben ser sincronizados con los modelos técnicos actuales o reemplazados por nuevos modelos técnicos (concepto tradicional).

5.2.2. Ejecución de modelos de negocio - ¿Es posible?

Toda persona que se ha interiorizado en el mundo de BPM, conoce el ciclo ideal de la gestión por procesos con apoyo de un Process Engine. Por lo general nos encontramos aquí con un grave problema: las expectativas que una BPM-Suite va a solucionar en forma mágica todos los problemas. La figura 5.3 muestra esta expectativa, o dicho de otra forma el ciclo deseado. La idea y el concepto que hay detrás de este deseo, es que la Suite se alimenta de un modelo de negocio (administrado por usuarios de negocio), integra en forma automática las diferentes soluciones de negocio y se encarga del Human-Workflow-Management. Finalmente la Suite entrega a los ejecutivos a través de un Dashboard indicadores de negocio, a través de los cuales el área de negocio puede resolver los problemas encontrados y hacer ellos mismos los cambios en tiempo real.

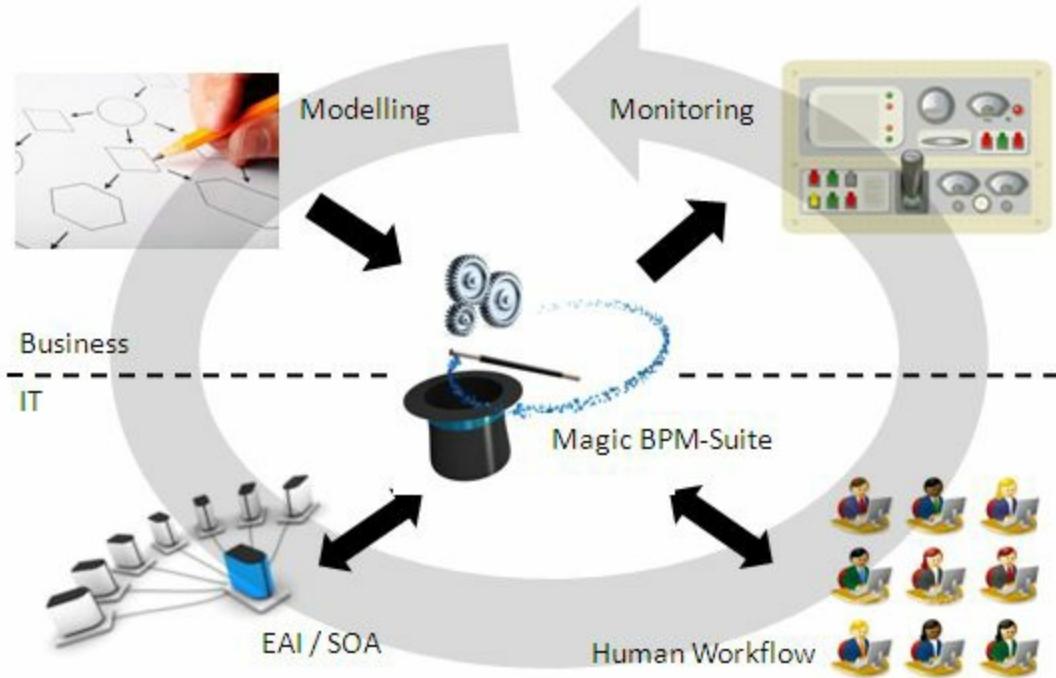


Figura 5.3: El

círculo mágico de la BPM-Suite

Este escenario suena demasiado bien para ser real, y así es en la práctica. Puede ser que una Suite mágica de BPM muestre la visión actual hacia donde se quiere llegar en el futuro, pero esta visión está aún lejos de ser realidad. Tenemos que advertir sobre las falsas expectativas que despiertan los proveedores tanto medianos como grandes de BPMS al respecto; el marketing les promete que existe esta Suite mágica. Muchos clientes y usuarios se dan cuenta, luego de haber adquirido estas plataformas, que sus proveedores no pueden cumplir estas expectativas, lo que se traduce en desilusión y en algunas veces en una oposición a la metodología de BPM.

Entonces ¿qué partes de esta visión pueden realmente funcionar y cuales no? ¿Pueden ejecutarse modelos de negocio? Si necesito una nueva capa de modelos técnicos, ¿cómo los sincronizo con los modelos de negocio? ¿No se habla tanto del mapeo automático entre modelos de BPMN y BPEL? Estas son la preguntas que queremos contestar a continuación.

Primero vamos a tratar si es posible ejecutar directamente modelos de negocio. Como explicamos en la sección anterior, existen realmente Process Engines que lo pueden hacer, pero el malentendido es que estos modelos los puedan desarrollar usuarios de negocio. Para que un modelo de negocio pueda automatizarse directamente, habrá que considerar muchos principios metodológicos y describir todo el detalle que se necesita para la implementación. Debemos recordar al lector, que un modelo de negocio ejecutable es sinónimo a código fuente de un lenguaje de programación. Un modelo de negocio, que se desarrolla para el negocio y no es automatizado, no requiere de tanto detalle, es más incluso molesta y se mantienen por general intencionalmente más agregados e incluso más informal, es decir con

márgenes de interpretación. Vamos enumerar algunos aspectos que son adicionalmente necesarios para implementar modelos de negocio:

Modelos de datos: Una especificación de los tipos de datos, no es necesario a nivel de modelo de negocio, pero si para la implementación de un modelo técnico. Al momento de la ejecución debe existir una definición técnica exacta de los tipos de datos que requiere la tecnología que se va a utilizar, por ejemplo en un esquema de XML. También el flujo de datos debe especificarse de acuerdo a la tecnología que la usa.

Casos de error: Si en el modelo de negocio sólo se consideraron errores propios del negocio, habrá que ampliar la casuística de posibles errores a errores técnicos que pueden ocurrir. Dependiendo de la arquitectura y de la tecnología, los errores técnicos pueden impactar en el modelo de negocio y vice versa.

Condiciones de correlación: Al igual que en el modelo de datos, habrá que definir con exactitud la correlación entre mensajería e instancias de procesos. El lector debe acordarse que el Process Engine administra múltiples instancias, por lo que esta debe saber qué mensajes corresponden a qué instancias en qué lugar y tiempo.

Instanciación de procesos: Iniciar un proceso técnicamente a través de eventos no es tan sencillo como lo muestra el diagrama, por ejemplo si un mensaje entrante en el modelo de negocio inicia un proceso, en la realidad este mensaje tiene que ir a buscarlo el Engine a algún lugar. O a lo mejor el inicio depende de la ocurrencia de múltiples eventos, que cada cierto tiempo el Engine tiene que revisar.

Estos ejemplos sirven para ilustrar por qué los modelos de negocio deben ser tan precisos y detallados si se quieren ejecutar directamente. Nuestra experiencia nos muestra que este grado de detalle no es adecuado para usuarios de negocio y no aportan para la comunicación entre ellos, razón por la cual en la realidad se hace necesario distinguir entre modelos técnicos y modelos de negocio.

Pero desde nuestro punto de vista esto no es tan grave como parece, porque desde el punto de vista de TI se recibe una representación gráfica de la lógica de negocio, que con pequeñas modificaciones y ampliaciones los ingenieros pueden implementar en la Suite de BPM. Justamente uno de los objetivos de BPMN es que no exista un quiebre de notación al pasar a un modelo técnico y que se pueda seguir trabajando con el mismo esquema, pero más técnico. Si se utiliza la misma notación para el modelo técnico facilita la comunicación entre la capa de negocio y la de TI, a pesar que ya no sigue siendo exactamente el mismo modelo.

Podemos resumir que se necesitan diferentes modelos, aunque sobre la misma notación, pero como puede percibirse el lector, esta expectativa ya no la cumple la Suite mágica de BPM.

Veamos ahora todos los enfoques que se han intentado desarrollar para cerrar esta brecha. Una idea de los ingenieros ha sido de mapear o relacionar los modelos, o mejor aún que de un modelo se pueda generar otro. En el enfoque del «Forward Engineering» se pretende que se puedan generar a partir de los modelos de negocio los modelos técnicos. Esto puede funcionar en una primera iteración, pero ya en una segunda iteración cuando se hayan introducido cambios en el modelo de negocio o en el técnico, nos encontramos con problemas que hasta el día de hoy aún no se han solucionados. Si hacemos un cambio en el modelo de negocio y volvemos a generar el modelo técnico, impacta masivamente sobre el actual. Se ha intentado proteger algunos bloques de código cuando se repite generación de modelos técnicos, pero en la práctica ha resultado más eficiente hacer las adaptaciones en forma manual. Por el otro lado si se realizan cambios en el modelo técnico no se hacen visibles en el modelo de negocio, o se debe hacer una adaptación manual.

Otro enfoque que se ha intentado realizar es el principio del «Roundtrip Engineering», en el cual los cambios se hacen primero en el modelo técnico y luego se genera automáticamente a una actualización del modelo de negocio. Actualmente este procedimiento se recomienda para el mapeo entre BPMN y BPEL, pero en proyectos reales no ha funcionado.

Pero estos problemas no sólo existen en el contexto de proyectos de BPM, sino también en el desarrollo de software, por ejemplo en el enfoque de «Model Driven Architecture (MDA)» también administrado por la OMG. Es interesante de notar que ambos enfoques (BPEL y MDA) hasta la fecha no han tenido éxito en proyectos reales. ¿Por qué? Desde nuestro punto de vista este problema tiene mucha relación con el paradigma del nivel de granularidad en los modelos: ¿Hasta qué nivel de profundidad podemos modelar sin programar?

Nuestro tip en BPMN: Business-IT-Alignment

Acepte que los modelos de negocio no son ejecutables directamente. Los intentos de generar modelos técnicos a partir de los modelos de negocio tampoco ha funcionado y los que han funcionado a modo de prototipo no han traído beneficios en proyectos reales. Un real beneficio lo vemos en que tanto los usuarios de negocio como los expertos de TI, trabajen con la misma técnica de diagramación. BPMN nos da esta posibilidad, ya que puede representar tanto modelos de negocio como modelos técnicos. La generación de código BPEL a partir de modelos de negocio desarrollados en BPMN ya no tiene beneficios.

5.3. Automatización de Procesos con BPMN 2.0

Una de las mayores novedades que trae la nueva versión 2.0 de BPMN es la introducción de la definición de una «semántica de ejecución» como también un «formato de serialización de XML». ¿Qué significado tienen estas expresiones? Esto se lo podemos explicar rápidamente: los modelos de BPMN pueden almacenarse en

un archivo de XML, pero la especificación norma como hacerlo. Existen dos tipos de esquemas de XML:

Para el intercambio de modelos: El XML contiene toda la información que se necesita para exportar e importar modelos de una herramienta a otra. La definición incluye información gráfica sobre el posicionamiento de los objetos (layout).

Para la ejecución de la semántica: La especificación norma cómo deben archivarse todos los detalles técnicos del proceso.

La especificación norma cómo deben ser almacenados los esquemas de XML para que se pueda interpretar la sintáctica, pero también la interpretación de la semántica y el metamodelo. Todo esto no existía en la versión anterior. Esta definición exacta permite que modelos de BPMN 2.0 puedan ser intercambiados, sin ampliaciones propietarias, intercambiadas entre herramientas que cumplen con la especificación, pero también que puedan ser ejecutadas con un Process Engine que interprete el código BPMN 2.0. Hasta antes de BPMN 2.0 éste era sólo un dominio del estándar Business Process Execution Language (BPEL). Debido a esta razón existía la idea de generar modelos de BPMN a BPEL.

Ahora la situación cambió y existen dos vías alternativas, mapeo a BPEL o ejecución directa de BPMN 2.0, pero no es objetivo de este libro tratar estos temas técnicos extensivamente en este libro, requeriríamos de uno propio. Al lector que se interese por el tema técnico Process Execution con BPMN 2.0, lo invitamos a conocer ejemplos en nuestro Blog (www.bpm-guide.de), o en la plataforma de Camunda BPM Open Source (<http://camunda.org/>) o bien analizar detenidamente en la especificación de BPMN el anexo «Documento oficial de ejemplo de Process Execution con BPMN 2.0».

5.3.1. El modelo de procesos técnico

Volviendo a nuestro ejemplo de la licitación de cargo para contratación de personal, tomamos como input nuestro modelo de negocio del nivel 2 (ver figura 4.13), pero consideramos sólo el pool del Process Engine. Para traspasar este modelo al nivel 3, hicimos unas pequeñas modificaciones (ver figura 5.4). El subprocesso «Realizar licitación» lo abrimos para mostrar la diferencia al modelo de negocio, pero en un proyecto real lo habríamos mantenido como subprocesso con el fin de no desviarnos del modelo de negocio si no es necesario. La licitación hay que realizarla en diferentes plataformas, situación que tenemos que reflejar en el modelo técnico. Tenemos que además indicar exactamente en qué plataformas se van a realizar estas licitaciones y con la finalidad pedagógica de mostrar qué más se necesita incluimos a modo de ejemplo un pool con un mensaje a una de estas plataformas. El mensaje como es representado aquí técnicamente no nos sirve y lo vamos a tratar más

adelante. En la realidad deberíamos incluir un pool por cada plataforma indicada en el lane del pool del Process Engine.

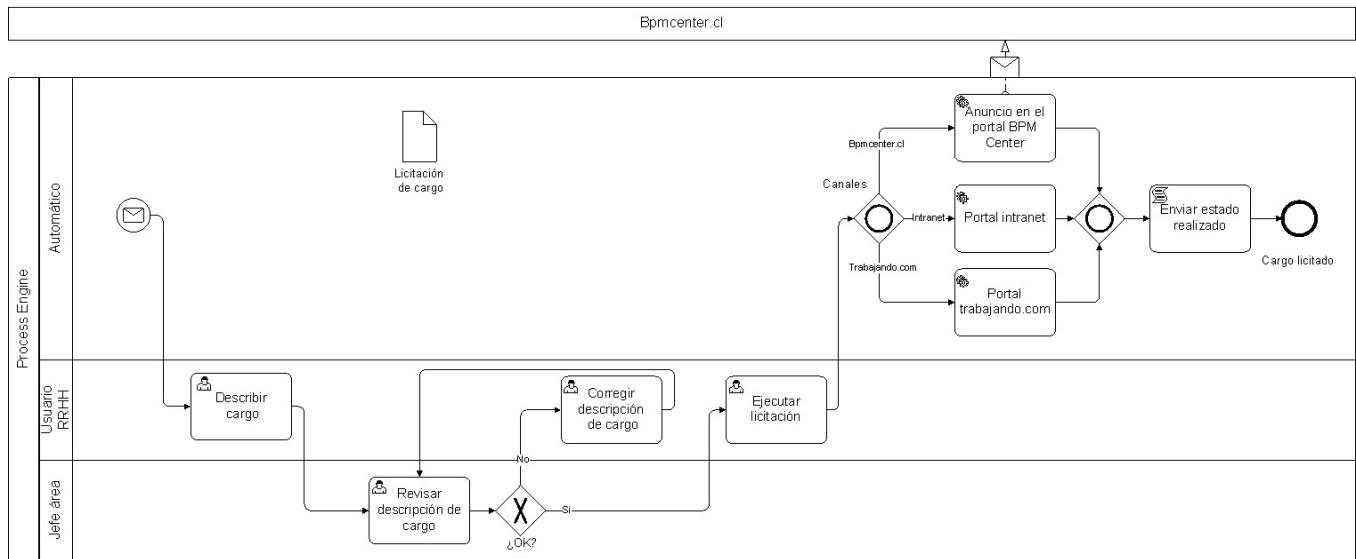


Figura 5.4: Modelo de proceso de licitación de cargo para nivel técnico

El proceso que usted aprecia en la figura 5.4 aún no muestra grandes diferencias con el modelo de la figura 4.13. En este caso las especificaciones técnicas se encuentran, por decirlo de forma vulgar «debajo del capó». Detrás de cada actividad del modelo encontrará un archivo XML con la especificación técnica en detalle. Como dicho anteriormente queremos recalcar algunos aspectos importantes para el modelo técnico, pero el lector que se interese por los detalles del formato XML puede consultar las páginas indicadas más arriba.

El ejemplo de la figura 5.4 corresponde en gran parte con el modelo del proceso de negocio. Las diferencias son las siguientes:

Incluimos un objeto de datos llamado «licitación de cargo», porque necesitamos almacenar ciertos datos en las instancias de los procesos.

Cambiamos el tipo de actividad «Enviar estado realizado» del tipo «envío» al tipo «script». La razón es que el envío puede suceder de diferentes formas, probablemente como e-mail. El e-mail podría enviarse invocando un servicio que pone a disposición la empresa, o bien directamente a través del Process Engine. Nosotros nos decidimos por el envío directo y para estos fines la Process Engine exige hacerlo a través de una actividad tipo script. Esta desviación al modelo de negocio debería de actualizarse en el modelo de negocio con el objetivo de mantener los modelos consistentes.

Estos son sólo algunos ejemplos de situaciones e iteraciones que suceden en la realidad cuando pasamos del nivel 2 al 3 y tenemos que hacer el diseño del modelo técnico.

5.3.2. Modelamiento de datos y expressions

En el proceso de nuestro ejemplo incluimos el objeto de datos «Licitación de cargo». Adicionalmente al objeto de datos en el diagrama se debe especificar el objeto de datos en el modelo técnico. BPMN no norma como debe especificarse el tipo de dato, pero ofrece puntos de intersección para integrar diferentes tipos de tecnologías. La preconfiguración estándar es un esquema de XML, pero también se pueden utilizar por ejemplo tipos de datos para Java o .Net.

Veamos la definición de datos de nuestro ejemplo como código XML. Utilizamos para estos fines el esquema XML «type language», importamos un archivo con el esquema de los tipos de datos para definirlos(«ausschreibungsDef»), el cual referencia a un elemento de XML en el esquema. Este tipo se utiliza en el objeto de datos como una variable del proceso:

```
<definitions ...
  typeLanguage="http://www.w3.org/2001/XMLSchema"
  expressionLanguage="http://www.w3.org/1999/XPath"
  xmlns:sample="http://sample.bpmn.camunda.com/"> ...
<import namespace="http://sample.bpmn.camunda.com/"

  location="SampleService.xsd"
  importType="http://www.w3.org/2001/XMLSchema" />

<itemDefinition id="ausschreibungDef" itemKind="Information"
  structureRef="sample:ausschreibung" />

...
<process id="Stellenausschreibungsprozess">
<dataObject id="ausschreibungVariable" name="Stellenausschreibung"
  itemSubjectRef="ausschreibungDef" />
```

Nota: Dejamos la expresión original alemán, por si el lector quiere seguir los ejemplos en los sitios indicados.
Traducción de las variables del alemán: «Ausschreibung = licitación», «Stellenausschreibungsprozess = proceso de licitación de cargo»,
«Stellenauschreibung = licitación de cargo»

BPMN diferencia entre objetos de datos que están representados en forma gráfica en el diagrama y propiedades que nos son visibles en el diagrama. En este último también se pueden declarar variables del proceso.

La especificación también ofrece lenguajes formales, por ejemplo para representar condiciones, lenguaje llamado «Expression Language». En este lenguaje las expresiones se denominan «Expressions». Como mencionamos en la introducción a

este capítulo, sólo utilizamos los términos originales en inglés.

De las Expressions se puede extraer nueva información de los datos actuales. En el caso más sencillo, sólo se evalúa si la expresión es «verdadera» o «falsa». Un buen ejemplo es la utilización XOR-Split: dependiendo de los datos en el proceso, el token sigue el flujo de unas de las alternativas del Gateway.

Por defecto se utiliza en BPMN el lenguaje XPath Expression, como se puede revisar en el código de XML. Xpath es un lenguaje de consulta que opera directamente sobre los datos de XML. Pero también este lenguaje puede ser cambiado por otro, por ejemplo por «Java Expression Language».

La libre configuración de tipos de datos en BPMN y la respectiva selección de la Expression Language, conlleva muchos grados de libertad para trabajar con diferentes tipos de Process Engine.

5.3.3. Invocación de servicios - ¿síncrono o asíncrono?

Bajo el término de asíncrono se entiende, que la respuesta de la llamada de un servicio no responde en un tiempo determinado, de forma que el servicio no se bloquea y puede seguir procesándose. En el modo síncrono es al revés, el servicio emitente espera la respuesta, que por lo general es instantánea o de corto lapso.

En los modelos técnicos de BPMN se diferencia si la comunicación es síncrona o asíncrona. La diferencia de ambas modalidades la vemos en la figura 5.5:

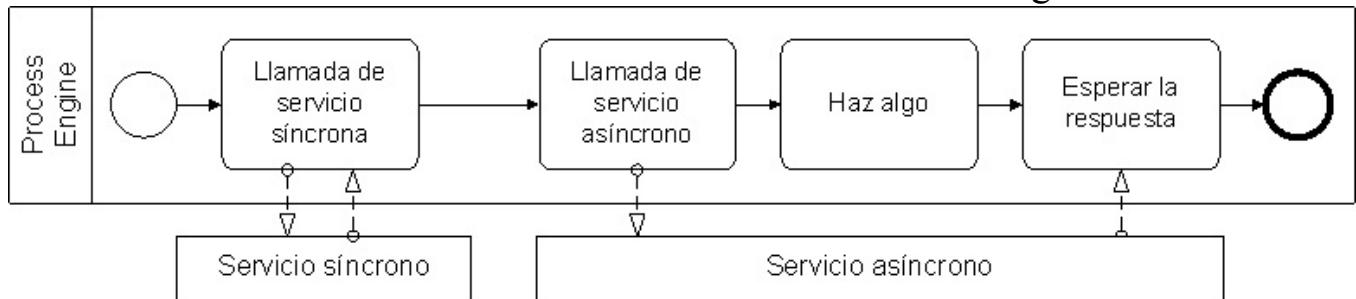


Figura 5.5: Llamada de servicios síncrono y asíncrono en BPMN

Para representar el caso asíncrono se pueden utilizar en BPMN alternativamente eventos de mensajes o actividades directamente, pero no en el caso del modo síncrono, porque los eventos pueden ser del tipo de captura o gatilladores, pero nunca ambos casos a la vez.

Si pensamos en categorías de desarrollo de software el caso síncrono es simple, porque técnicamente se espera una respuesta inmediata. Se podría comparar con la llamada de una función con variable de retorno en un lenguaje de programación. En el caso asíncrono es un tanto más complicado porque la respuesta se espera más tarde, entonces se necesita sincronizar el llamado con la respectiva instancia del

proceso, lo que se soluciona con el concepto de correlación. La condición de correlación debe estar por consiguiente bien definida para el proceso de sincronización.

En discusiones de proyectos reales nos encontramos con la situación que existen casos que el modelo de negocio indica un llamado asíncrono, pero que técnicamente se implementa como un caso síncrono. En nuestro ejemplo de la pizzería el lector puede apreciar en la figura 5.6 esta situación. Supongamos que queremos pedir una pizza y llamamos a la pizzería para hacer el pedido.

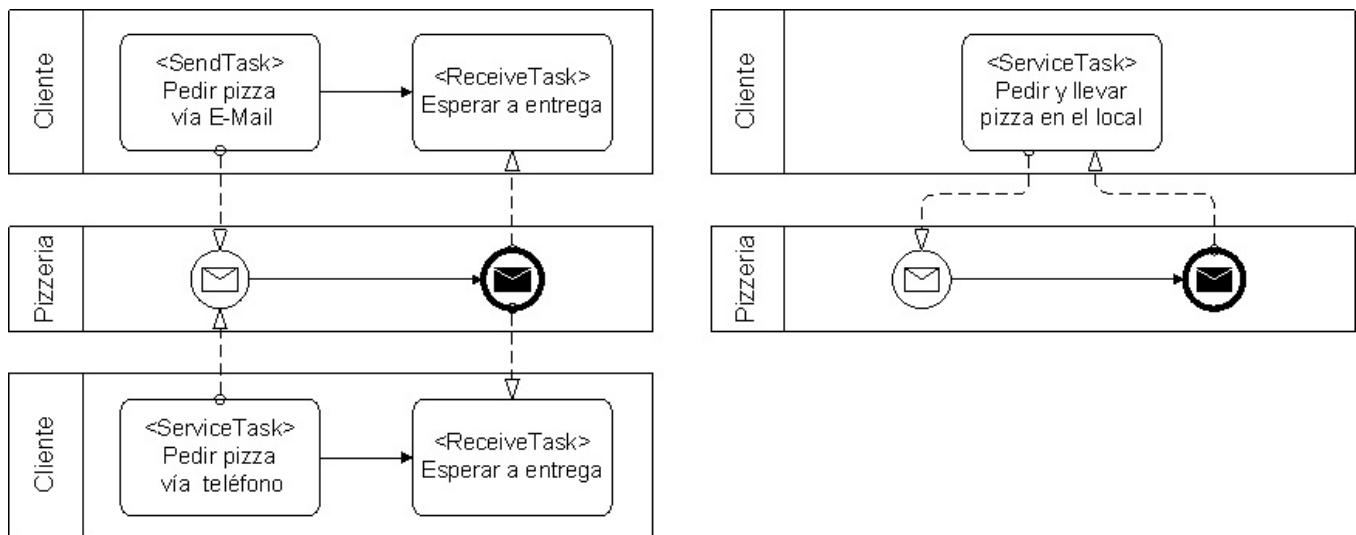


Figura 5.6: El modo asíncrono o síncrono no es siempre igual en el nivel de negocio y el técnico

Técnicamente podríamos decir que se trata de un llamado síncrono porque la línea telefónica se mantiene hasta que nos tomen el pedido, pero el servicio de la pizza se realiza en forma asíncrona, porque la entrega será tiempo más tarde.

Alternativamente podríamos enviar un e-mail para hacer el pedido, desde el punto de vista técnico este sería un caso asíncrono, porque no sabemos si la pizzería lo recibió, o si fue leído.

Otra alternativa sería que fuéramos personalmente a la pizzería, hicieramos el pedido en el local y esperáramos hasta que estuviera lista. Este caso corresponde a un llamado síncrono, porque no dejamos el lugar hasta que nos entreguen el pedido. Entretanto no podemos hacer ninguna otra cosa, similar a un proceso que espera la respuesta.

El lector se preguntará ¿por qué insistimos tanto en este juego que el caso sea de una u otra forma? Porque en la realidad nos vamos a encontrar con estas situaciones.

Nuestro Tip en BPMN: Business-IT-Alignment

La vista técnica y de negocio en cuanto a situaciones síncronas o asíncronas pueden ser diferentes. Téngalo presente y tome conciencia al respecto.

5.3.4. Integración de sistemas

La siguiente pregunta que hay que responder con respecto al modelo técnico, es cómo BPMN trata la integración de sistemas. La notación dispone como en otros aspectos también de una configuración estándar para invocar sistemas, que son llamados de webservices, pero habrá que tomar la decisión si se requiere de una integración síncrona o asíncrona. Para llamar un servicio web hay que definir un mensaje de entrada o de salida con los respectivos parámetros. BPMN dispone de una definición técnica para la configuración de mensajería de servicios, pero también existen posibilidades de integrar otro tipo de tecnologías como Java o .Net.

Por otra parte existe la situación que otros sistemas inician las instancias de un proceso, en el modelo representado gráficamente por eventos de inicio. BPMN provee para estos casos también una interfaz del tipo servicio web.

5.3.5. Actividad de usuario

El último aspecto a considerar en el modelo técnico es cómo configurar la interacción humana (Human Interaction) en una actividad del tipo usuario (User Task). Las actividades de usuario o tareas en el proceso, son técnicamente administradas a través de un listado de actividades. El usuario recibe un listado de tareas. Si el usuario selecciona una de ellas, el proceso se activa y el proceso continúa hasta cuando el usuario haya confirmado que terminó de procesarla.

BPMN conoce tres formas técnicas de integrar la administración de tareas: Se podría utilizar un servicio web genérico, que se integra a la administración de tareas propietaria del Process Engine.

Se podría integrar directamente una funcionalidad propietaria del Process Engine, por ejemplo en Java. Esta segunda posibilidad puede ser más fácil y rápida de integrar, pero por otro lado la arquitectura pierde independencia al entregar una funcionalidad importante a una tecnología propietaria.

La tercera posibilidad es de utilizar una definición estándar de la notación muy completa llamada «WS-Human-Task (WS-HT)». Esta interfaz está muy bien documentada en todo detalle y es muy poderosa. Considera aspectos como roles, delegación, escalación e incluso meta-information para configurar la forma de presentación, por ejemplo el campo de referencia. Esta interfaz es aún muy nueva y los fabricantes la están integrando en sus sistemas lentamente. Si una Process Engine ofrece la alternativa de integrar la WS-HT al administrador de tareas, es la mejor

opción de utilizarla.

5.4. Semántica de Ejecución

Como lo habíamos dicho anteriormente, no es objetivo de este libro explicar en detalle toda la especificación técnica de BPMN 2.0, pero queremos presentarle en forma resumida los aspectos más importantes de la semántica de ejecución.

5.4.1. Eventos de inicio e instancias de procesos

Los eventos de inicio crean nuevas instancias de procesos, pero ¿cómo lo hacen realmente? Si usted es una persona de TI seguramente estará pensando en dos posibilidades:

La nueva instancia la crea el proceso mismo a través de alguna acción con apoyo del Process Engine.

La nueva instancia es creada por una componente externa de TI.

Para estos efectos supongamos el siguiente ejemplo: Se crea una nueva instancia de proceso, cuando recepcionamos un e-mail. Desde el punto de vista técnico, este e-mail de alguna forma hay que ir a buscarlo, interpretar los datos contenidos en él y luego iniciar la instancia del proceso. Lógicamente que este caso lo podemos modelar como muestra la figura 5.7. Si el Process Engine apoya esta funcionalidad, estaría minuto a minuto consultando el administrador de correos.

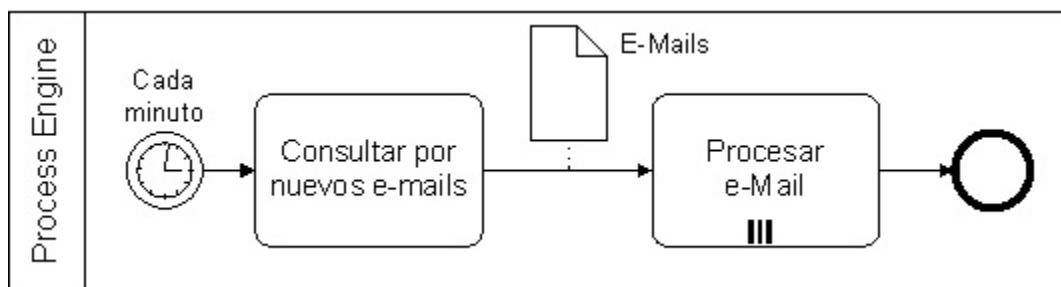


Figura 5.7:
Modelo para creación de una instancia de proceso con evento de tiempo

Dependiendo de los servicios generales que tenga en una empresa también es posible que existan componentes de TI, que ofrezcan esta funcionalidad, como lo muestra la figura 5.8. Por lo general un Enterprise Service Bus (ESB) se encarga de entregar este servicio.

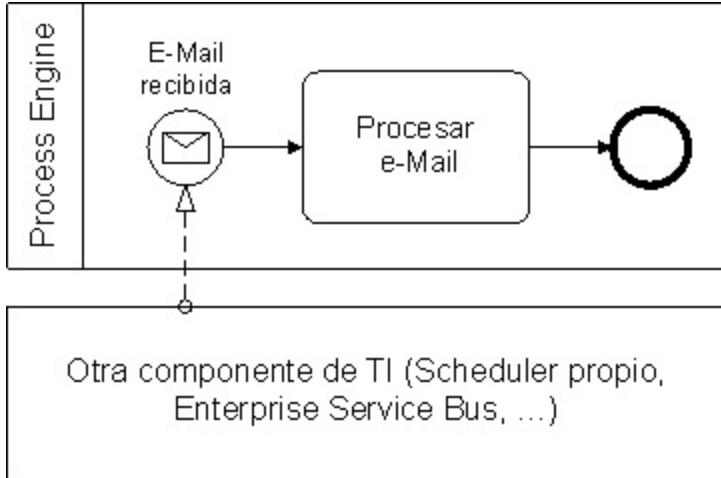


Figura 5.8: Modelo para creación de una instancia de Proceso impulsada por componente de TI externa

En nuestros proyectos reales nos encontramos con ambas situaciones. Por lo general si una empresa cuenta con una arquitectura de SOA centralizada, nos encontramos con muchos servicios globales (como en figura 5.8) y en empresas medianas o pequeñas el Process Engine debe internalizar todas estas funciones (como en figura 5.7).

Ambas alternativas tienen ventajas y desventajas. Los modelos de negocio son más entendibles, si no cuenta con tantas interfaces, pero desde el punto de vista técnico se liberan muchas configuraciones propietarias si se cuenta con servicios centralizados. Podemos resumir que siempre dependemos de la arquitectura de TI existente.

Eventos de inicio múltiples

En la sección 2.6 explicamos la situación que desde el punto de vista de negocio tienen que suceder varios eventos para que se cumpla la condición de iniciar la instancia de un proceso. Este caso es técnicamente más complejo, de lo que indica el modelo de negocio, debido a que la semántica debe estar definida en forma exacta desde el punto de vista de TI.

Un ejemplo lo encuentra en la figura 5.9: Un Broker (Agente en la bolsa de comercio) recibe solicitudes de compra y venta en la bolsa de comercio. En este caso de estudio vamos a suponer que los compradores y los vendedores no se conocen y el Broker transa acciones en el mercado de valores. Si entra primero una orden de compra (OC), entonces el Token de arriba va a avanzar hasta el AND-Join, pero la instancia de procesos no puede terminar si no existe una correspondiente orden de venta. La gracia que tiene el Parallel-Event-Gateway es que puede relacionar (relacionar en el sentido de correlación de llaves) estos eventos. En nuestro caso, si entra una oferta de venta, el Gateway puede relacionar si corresponde con la oferta de compra. Si corresponde sigue el flujo, de lo contrario

se crea una nueva instancia.

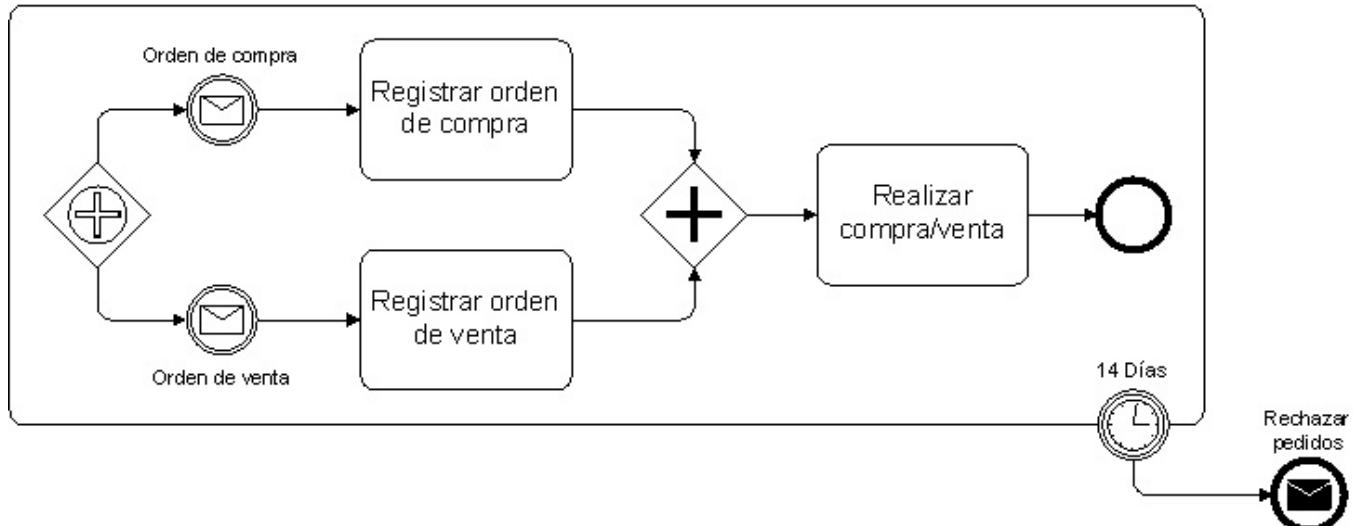


Figura 5.9: Ejemplo de un Parallel-Event-Gateway como inicio múltiple

Desde el punto de vista del Process Engine, existe el grave problema que si no entra una oferta de venta que corresponda con la de compra, la instancia podría morirse de hambre esperando. Lógicamente que este caso hay que detenerlo con un time-out.

La figura 5.10 muestra en forma genérica diferentes patrones para el caso de eventos de inicio múltiple. Nuestro ejemplo cae en el caso (d). El caso (c) en el modelo de negocio es sinónimo al de (d), pero para el modelo técnico el caso (c) no es aplicable, porque el AND-Join no puede correlar los eventos y a pesar que existiese una orden que calce con la otra, sin correlación técnicamente no se unirían nunca. Como puede apreciar el lector en estos ejemplos simples, en el modelo técnico no puede haber margen de interpretación.

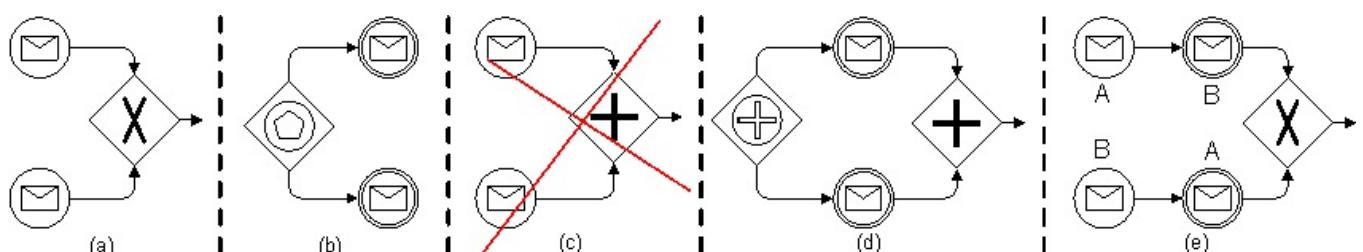


Figura 5.10: Diferentes patrones para caso de eventos de inicio múltiple

Los casos (a) y (b) representan el mismo caso. Entonces ¿por qué existen estos dos patrones si representan lo mismo? Algunos estudios han mostrado que algunas personas comprenden mejor modelos de procesos, si ven al inicio sólo un objeto de inicio. El Event-Gateway cumple con esta función visual.

Finalmente también el patrón mostrado en el caso (e) está permitido (Nota: este era la única forma de representar inicios por eventos múltiples en la versión 1.2), pero la correlación tiene que modelarse en forma externa, el Process Engine, no puede reconocerlo. Un caso de estudio teórico sobre estos patrones encontrará el lector en

[DM08].

5.4.2. Eventos intermedios y cómo se implementan en TI

Desde el punto de vista del modelo técnico eventos intermedios necesitan de un trato especial. Primero vamos a analizar el fenómeno que algunos eventos intermedios se «esfuman» cuando ocurren, como describimos en la sección 2.6.1. Este es el caso cuando ocurren eventos, pero ninguna instancia de procesos está esperándolos porque la actividad correspondiente ya fue procesada. En los modelos de negocio por lo general no se representan estos casos porque los diagramas se volverían muy complejos. Para representar este caso en el modelo técnico podríamos adicionar al evento intermedio un símbolo de cola (en inglés: queue) como artefacto.

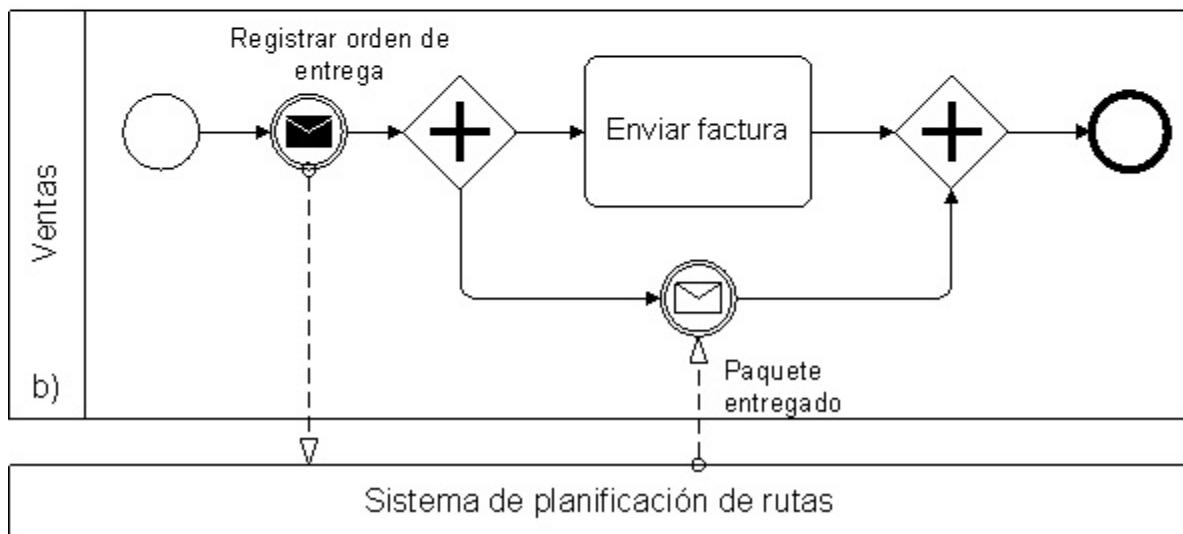
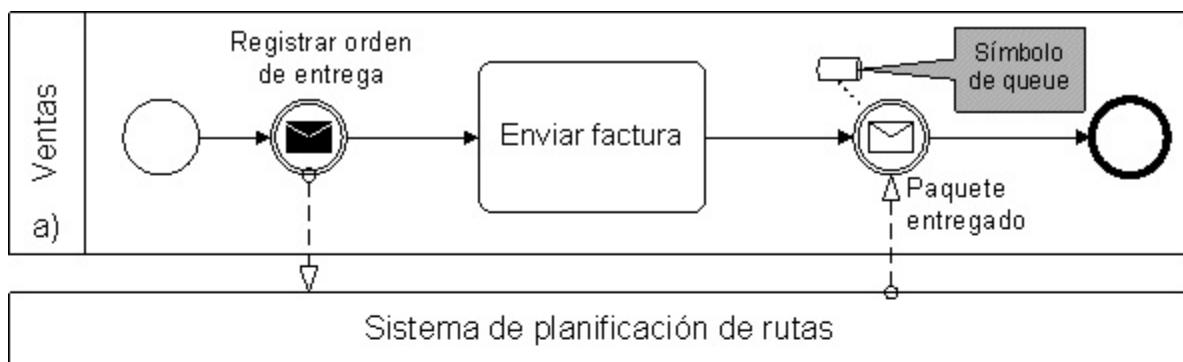


Figura 5.11: Eventos intermedios en el modelo técnico: caso a) con inconsistencia y caso b) con trato excepcional

El ejemplo en la figura 5.11 a) muestra este caso: Una orden de entrega se registra en el sistema de planificación de rutas y en seguida se emite la factura del servicio. Supongamos que el servicio de entrega tiene que llevar la factura, pero esta aún no ha salido a pesar que el paquete se encuentra en el vehículo, por lo que nos encontramos para este caso con una inconsistencia, lo que nos obliga a rediseñar en

esta parte el modelo de negocio. En este caso parallelizamos el flujo como lo muestra la figura 5.11 b) para solucionar el caso excepcional.

Analicemos ahora el caso de los eventos del tipo de condición. Para estos fines vamos a utilizar el ejemplo de la figura 5.12: Una orden de compra sólo puede confirmarse si el producto se encuentra disponible. Desde el punto de vista de negocio el modelo que el lector aprecia en la parte a) es suficiente, pero no para un modelo técnico, porque si lo dejáramos así el algunos casos la instancia nunca llegaría a su término. Pero además hay otros aspectos que considerar:

Especificación de la condición: En el Process Engine se debe especificar la condición formalmente. En BPMN se puede formular por medio de las Expressions, por default en XPath. Si se acuerda el lector en la sección 2.6.5 explicamos que la condición tiene que cumplirse en forma independiente del proceso. ¿Cómo compatibilizamos esto? ¿Cómo se puede evaluar esta condición entonces? En BPMN podemos acceder a través de XPath u otras Expression Languages a servicios externos. Debemos de reconocer que esta salida no es metódicamente correcta, pero es la forma como lo puede hacer.

Escurrimiento: En eventos del tipo de condición también puede ocurrir el fenómeno del escurrimiento (el evento se esfuma). Analicemos el caso de la figura 5.12: Desde el punto de vista de negocio el producto se entrega cuando esté disponible (caso a). ¿Qué pasa si el producto está disponible en el momento que el Token entra al evento de condición? Desde el punto de vista técnico la condición sucedió en el tiempo pasado, es decir para nuestra instancia de proceso se «escurrió». Esto lo podemos corregir modelando como lo muestra la figura 5.12 b).

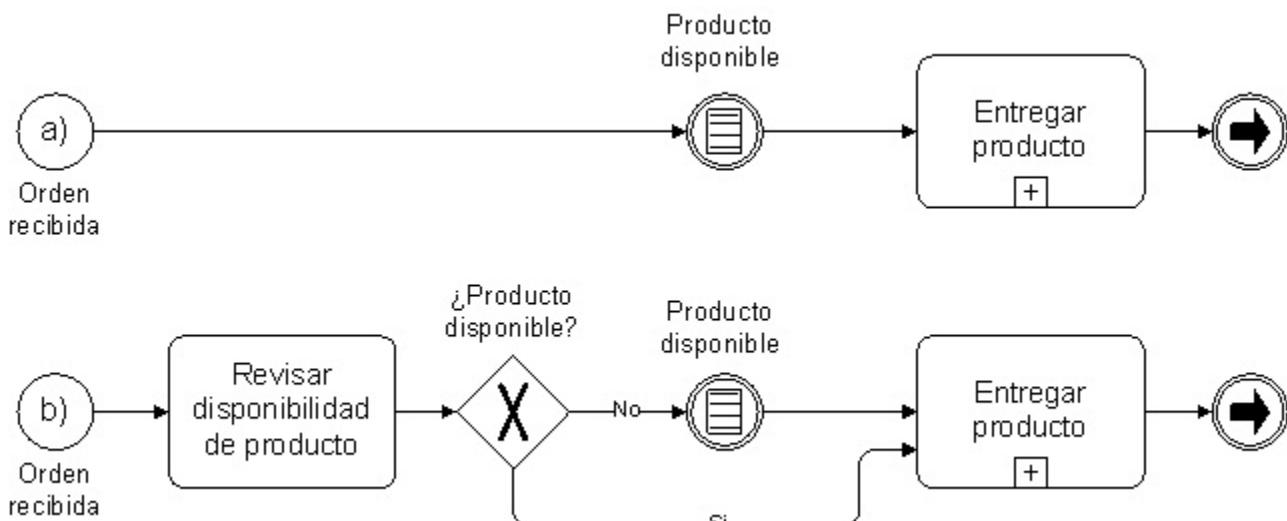


Figura 5.12: Utilización de evento intermedio de condición en a) modelo de negocio y b) modelo técnico

Alternativas para evaluar condiciones: Al igual que en el caso de iniciar una instancia de proceso, habrá que preguntarse desde el punto de vista técnico, quién se

encarga de la evaluación y cómo se lo hacemos saber al Process Engine. Como lo hemos dicho en reiteradas ocasiones, la especificación no considera en eventos funcionalidades de revisión, por lo que le mostramos dos alternativas de solución (ver figura 5.13):

- Caso a): Una componente de TI externa realiza la evaluación y le informa al proceso sobre los resultados. Este componente podría ser un ESB que se encarga de tomar esta funcionalidad. Entonces tenemos que preguntar como formular la Expression para este caso. Con la tecnología de Event Driven Architecture (EDA) existe un lenguaje de consulta para eventos, pero estas funcionalidades técnicas no son parte de la especificación de BPMN. A los proveedores de tecnología se les da libertad de como integrar estas funcionalidades con BPMN.
- Caso b): En este caso el Process Engine debe evaluar si la condición se cumple o no. Esto lo tiene que hacer cada cierto intervalo de tiempo. La desventaja de este procedimiento es que no es muy eficiente, le pide muchos recursos al Process Engine.

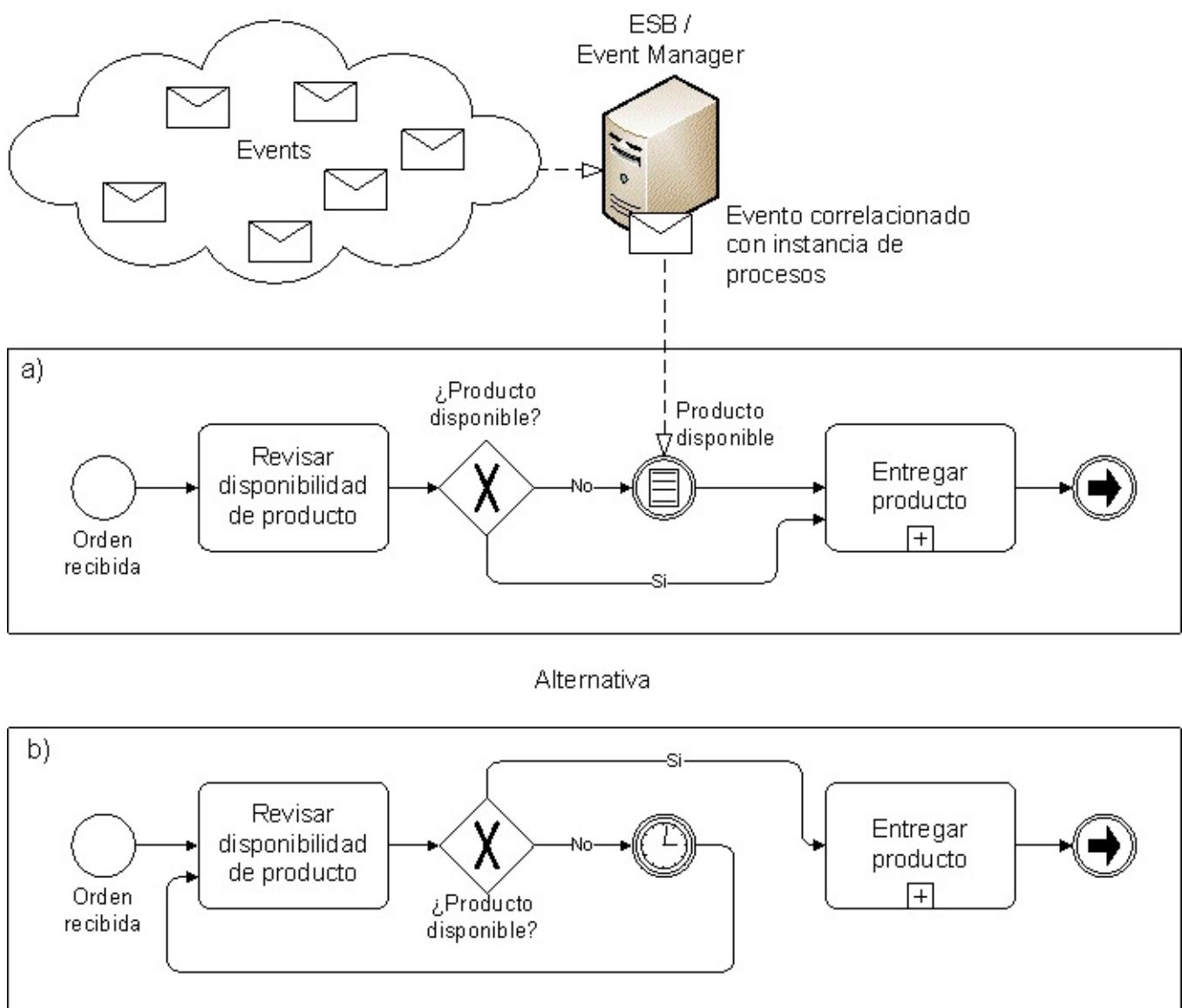


Figura 5.13: Evaluación de condiciones a) con implementación de tecnología externa y b) sin implementación de tecnología externa

5.4.3. Correlación

En la sección anterior explicamos que la correlación entre eventos de captura e instancias de procesos es importante en modelos técnicos. La pregunta técnica que ha de responderse es: ¿A qué instancia(s) de proceso(s) se le asignan los eventos entrantes (de captura)? La especificación de BPMN provee dos enfoques para solucionar la correlación:

Correlación de llaves (en inglés: key-based): Para cada instancia de comunicación (conversación) el Process Engine crea una llave artificial que debe estar contenida en cada llamado de servicio o mensajería. El Process Engine puede relacionar fácilmente las instancias correspondientes, pero este enfoque es muy rígido y cada participante se encuentra amarrado a la llave artificial.

Correlación de contexto (en inglés: context-based): El enfoque de correlación basado en el contexto es la alternativa más limpia y cerca del negocio. El número de la orden compra podría ser por ejemplo la llave del contexto. También se pueden declarar propiedades como variables del proceso o mensajes que se asocian a las instancias. Las declaraciones se hacen en las Expressions de BPMN.

Si la correlación se encuentra bien definida, el Process Engine se encarga del resto. La especificación habla de «Instance Routing», sin que sea necesario utilizar mayor infraestructura.

Pero a pesar de estas posibilidades de relacionar instancias con eventos, la correlación sigue siendo desde el punto de vista técnico un tema complejo. A continuación enumeramos algunos de los problemas que pueden suceder:

¿Qué pasa si un evento no se puede correlacionar? Este es el caso del escorrimiento, siempre ocurren situaciones que no se han previsto. Entonces la información del evento se registra en un administrador de cola y estos casos habrá que analizarlos en forma manual.

¿Qué pasa si un proceso cambia su estado? Una instancia de procesos puede haber dejado su estado esperado, por ejemplo que haya sido manipulado por el administrador o bien un time out. En estos casos los eventos quedan atascados en el administrador de colas, pero también existen casos peores como en el que el proceso se encuentra en un estado más avanzado y el evento entrante se correlaciona a una actividad que no le corresponde. Para solucionar estos casos poco deseados, habría que ampliar el concepto de correlación y por ejemplo incluir al registro de llaves los estados de las instancias. En este último caso se hace necesario ampliar los tipos de

datos en los sistemas que colaboran con el proceso.

Correlaciones condicionales falsas: Pero el caso es si suceden asignaciones falsas, es decir que se correlacionen eventos a instancias que no corresponden. El origen de estas faltas han de encontrarse en casos sofisticados que no han sido bien modelados. Este tipo de errores deberían evitarse con un buen testeo antes de salir a producción.

5.4.4. Gateways

En la sección 2.3 se explicaron los diferentes tipos de Gateways. BPMN 2.0 definió una semántica de ejecución bien precisa que le queremos explicar a continuación. Semántica de ejecución significa en este caso, como se consumen los Tokens, cuando nacen en Gateways nuevos Tokens, y por cuales flujos continúan estos.

Dos son los casos más sencillos como lo muestra la figura 5.14. El Gateway Paralelo (Fork and Join) asigna a un Token de todos los flujos de entrada y crea nuevos Tokens para cada una de las salidas. Se consume un Token por cada flujo. Si existen varios Token en un flujo (lo que en la práctica no tiene mucho sentido), esperan los Tokens restantes en el flujo a la próxima activación del Gateway.

En el caso del Gateway Exclusivo (XOR-Join and XOR-Split) se rutea exactamente cada Token entrante a un flujo saliente. Las condiciones indicadas en el flujo son evaluadas (declaradas en la Expression) en forma secuencial. El Token se rutea de acuerdo a la primera condición válida encontrada. El orden de ruteo lo determina el XML. Si ninguna condición es válida, se rutea al flujo por defecto y si no está declarado el flujo por defecto, el sistema lo lleva a una Exception, debido a que se detecta una situación inválida.



Figura 5.14: Ruteo técnico de Tokens aplicado a Gateways Paralelo y Exclusivo

El caso del Gateway Inclusivo es más complejo (ver figura 5.15). Analicemos primero el caso de un Token entrante (OR-Join). El Gateway se activa, cuando en todos los flujos entrantes se captén Tokens o bien no esperan otros Tokens. ¿No se esperan otros Tokens? ¡más fácil decirlo que realizarlo! Es un desafío para los fabricantes controlar la semántica de ejecución de Gateway Inclusivo. ¿Cómo saber

si se espera un Token o no? La condición de negocio dice que todo es posible, pero este tipo de condiciones son enemigos de la tecnología. Veamos el ejemplo de la figura 5.16: El OR-Split sabe recién después de haberse ejecutado la actividad 2, si el flujo de abajo puede activarse o no. Existen aún patrones más complicados con condiciones de loop, etc..

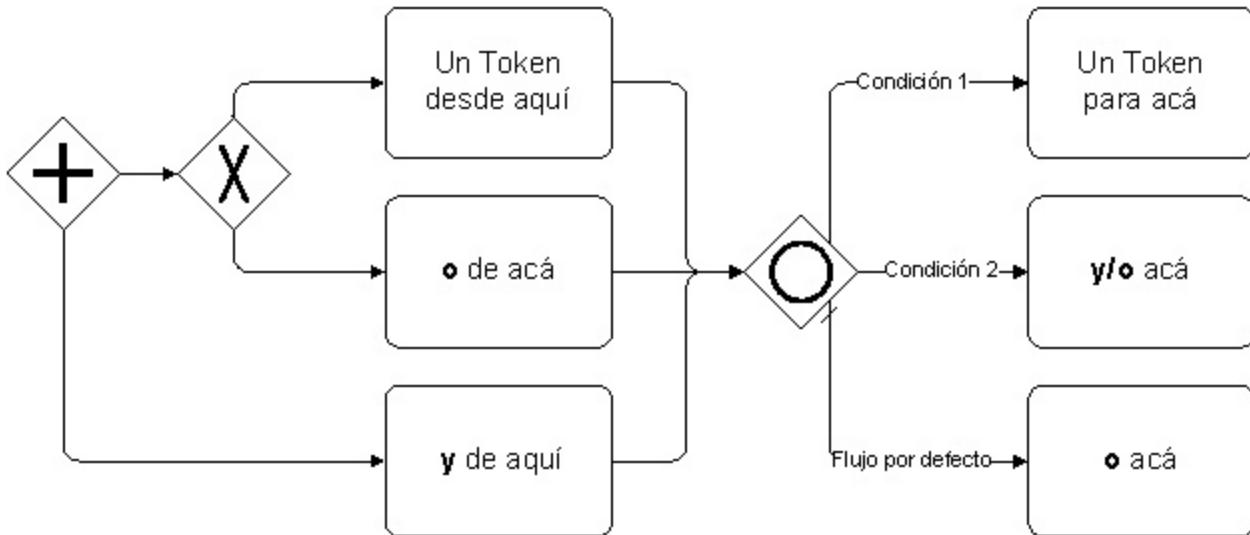


Figura 5.15: Gateway Inclusivo (OR)

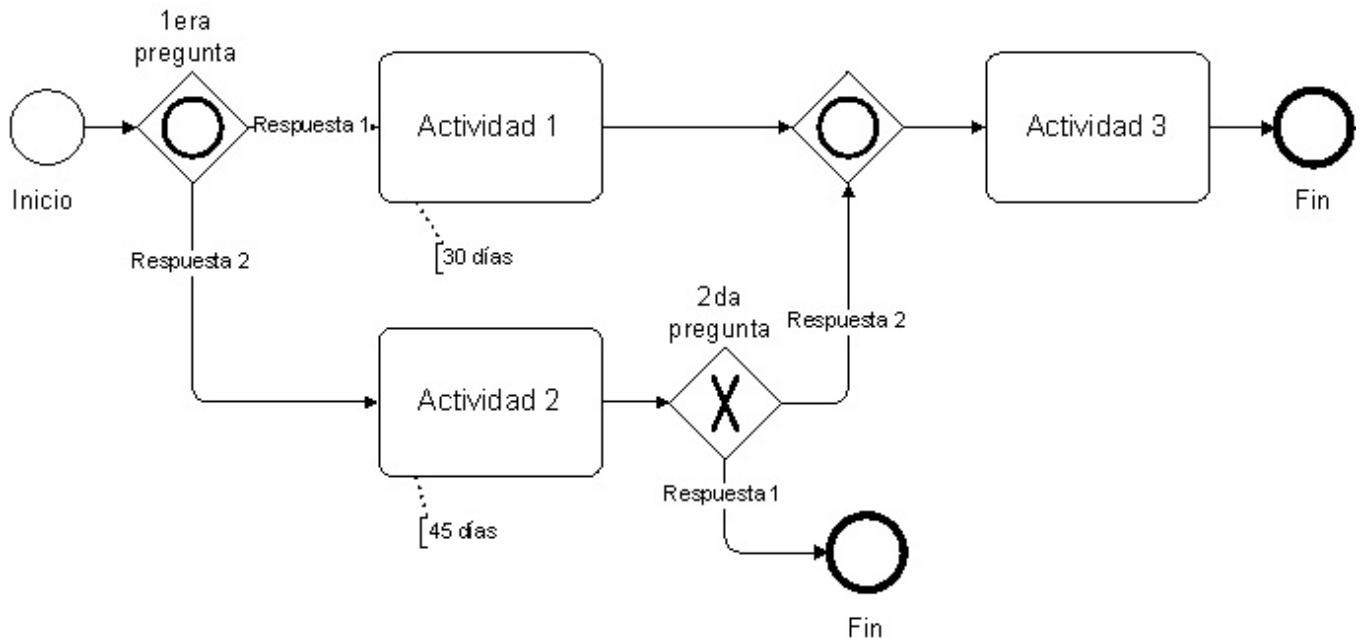


Figura 5.16: El comportamiento de espera del OR-Gateway

El caso del OR-Split es menos complicado, porque cada flujo de salida puede ser evaluado si cumple con la condición expresada. En el OR-Split nacen tantas instancias de Tokens como condiciones válidas se hayan encontrado. Si ninguna condición es válida, se deriva un Token al flujo por defecto y al igual que en el caso del XOR-Split y si no se ha declarado un flujo por defecto el sistema lo lleva a una Exception, debido a que se detecta una situación inválida.

El Gateway Complejo, es como su nombre lo indica «complejo». Puede responder a requerimientos especializados que no se pueden representar con los casos anteriores.

Pero ¿cómo se configura? Veamos el ejemplo de la figura 5.17: Un token entra por el flujo «a», o «b» y un tercero por «c». Si más tarde entra un Token por «d» no lo podemos captar. ¿Por qué? porque la tecnología de Gateway conoce sólo dos estados: «esperar a inicio» y «esperar a reset». El esperar a inicio está activo hasta que el Gateway se gatille y luego vuelve inmediatamente al estado de espera a reset, por lo que no puede captar en una sesión la entrada tardía de «d». Pero si la condición de negocio quiere esperar a «d», ¿cómo hacerlo?

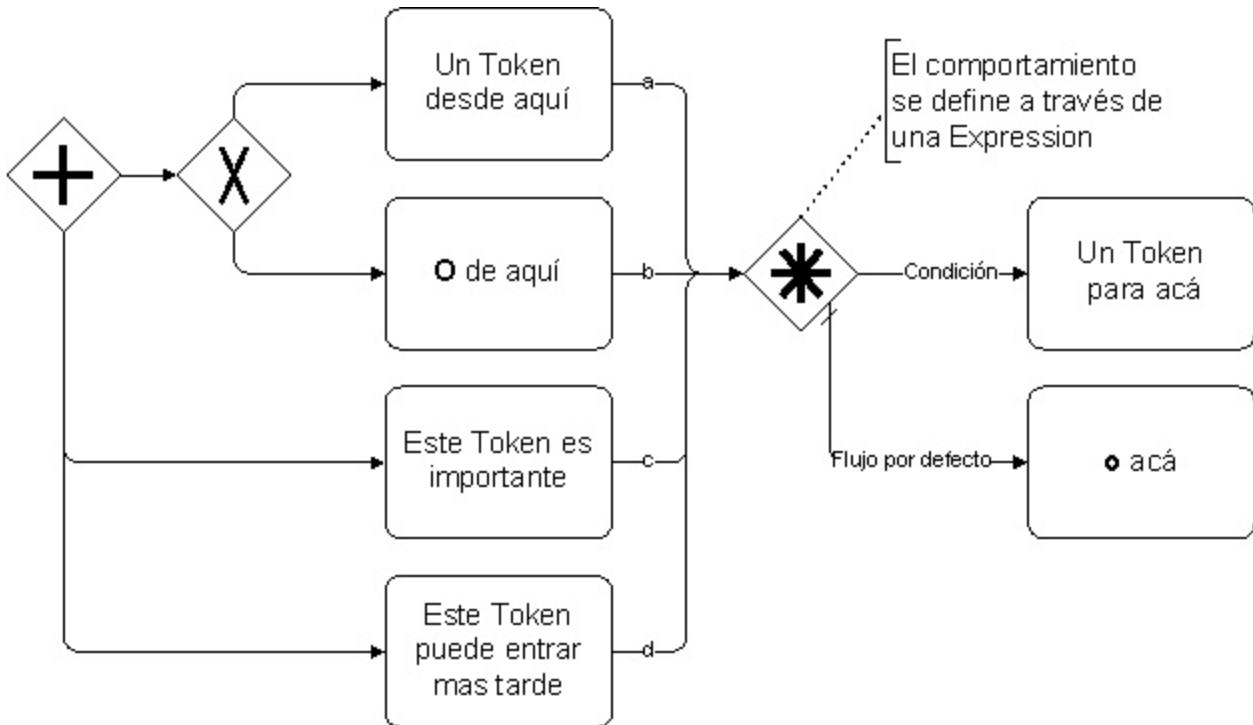


Figura 5.17: Gateway Complejo

El truco que emplea la especificación es que las condiciones del Gateway Complejo se especifican en una Expression. El Process Engine contabiliza todos los Tokens entrantes y está información queda disponible en la Expression, una gran diferencia al OR-Join. Entonces podemos expresar: « $(1 \times a \text{ o } 1 \times b) \text{ y } 1 \times c$ ». El comportamiento del Split en el Gateway complejo es idéntico al del OR-Split, por lo que no es necesario volver a explicarlo.

Finalmente el comportamiento técnico del Event-Gateway: En el queda esperando el Token entrante hasta que ocurra un evento que cumpla con una de las condiciones de salida, entonces el Token va a salir exactamente por ese flujo.

5.4.5. Finalizar una instancia de proceso

Una instancia de proceso puede finalizar bajo diferentes circunstancias. Por un lado puede terminar en forma «normal», pero dependiendo del flujo puede llegar a términos anormales como son término con interrupción por «error», «escalación» u otros.

Analicemos primero el caso normal: Una instancia de proceso se considera

terminada (en inglés: completed) cuando todos los Tokens hayan llegado a su estado de término. Un estado de término puede estar relacionado con un evento de término o se puede interpretar que una actividad que no tenga un flujo de salida es una actividad «final».

Nota: A excepción de los subprocessos, no recomendamos dejar actividades finales sin eventos de término porque se podría confundir con un error sintáctico.

La figura 5.18 muestra términos normales en varios flujos con eventos de término indefinidos. Para que en este ejemplo termine la instancia del proceso, tienen que haberse consumido todos los Tokens pA excepción de los subprocessos, no recomendamos dejar actividades finales sin eventos de término porque se podría confundir con un error sintáctico provenientes de los flujos paralelos.

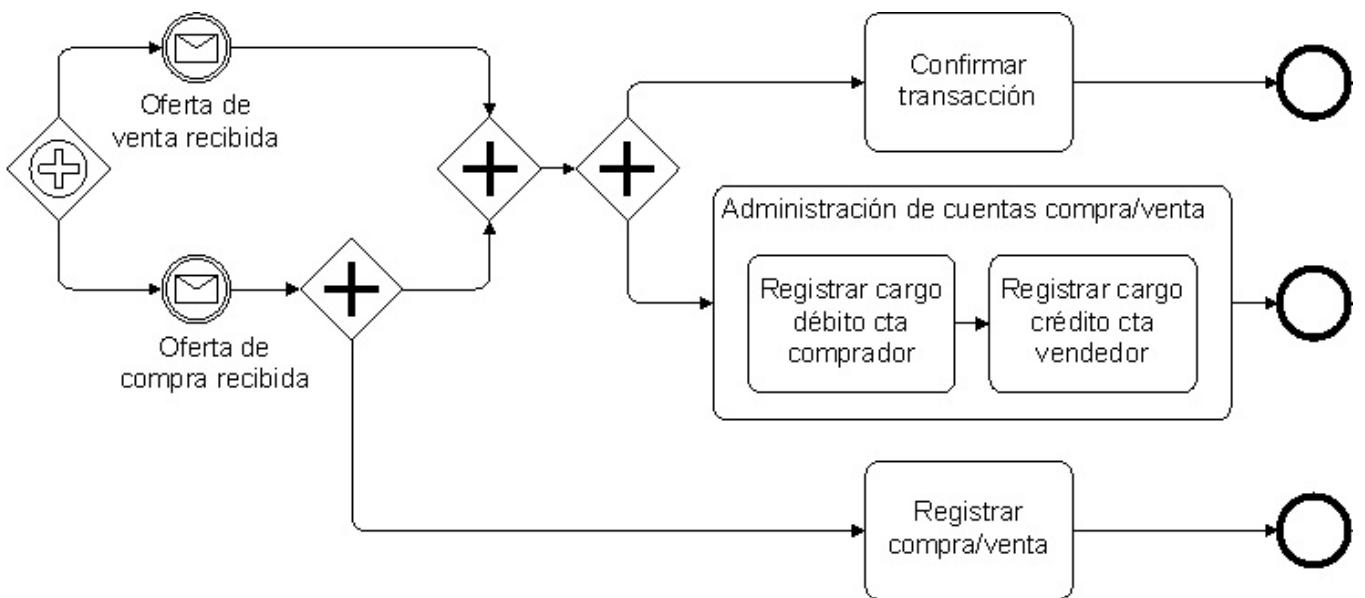


Figura 5.18: Caso de proceso que se inicia con Event-Parallel-Gateway y finaliza con evento de término normal

El ejemplo muestra también el tratamiento de un subprocesso con eventos de término. La última actividad del subprocesso carece de un flujo de secuencia de salida, razón por la cual se interpreta como actividad final del subprocesso. Los subprocessos tienen con respecto a los eventos finales una característica especial, terminan por sí «sólos» y transmiten siempre su término al proceso superior que lo controla. Visto de otra forma un proceso superior no puede terminar, si no han finalizados todos sus subprocessos.

Existen diferentes eventos de términos los cuales podemos clasificar en normales y anormales (ver figura 5.19). A los eventos normales se les puede definir una acción antes de terminar:

Enviar mensaje (Message end event)

Enviar señal (Signal end event) Término con compensación (Compensation end

event)

Dentro de los eventos de términos anormales conocemos:

Eventos terminales (termination end event): Todos los Tokens activos se finalizan de inmediato, inclusive la instancia completa.

Término por error (Error end event): Igual que en el caso anterior se finalizan todos los tokens y se transmite un mensaje de error al proceso superior. Si no existe un proceso superior, o si este no reacciona ante el mensaje, la especificación deja el tratamiento abierto.

Término por cancelación (Cancel end event): Igual que en el caso anterior (tratamiento de error), pero adicionalmente se devuelve la transacción y se ejecutan tareas de compensación si están indicadas.

Término por escalación (Escalation end event): El término por escalación sólo puede estar relacionado con un proceso superior. El resto de los Token activos no se terminan, a no ser que el subprocesso haya activado un token intermedio de interrupción junto al de escalación. Sólo en este caso se finalizan todos los Tokens activos del subprocesso.

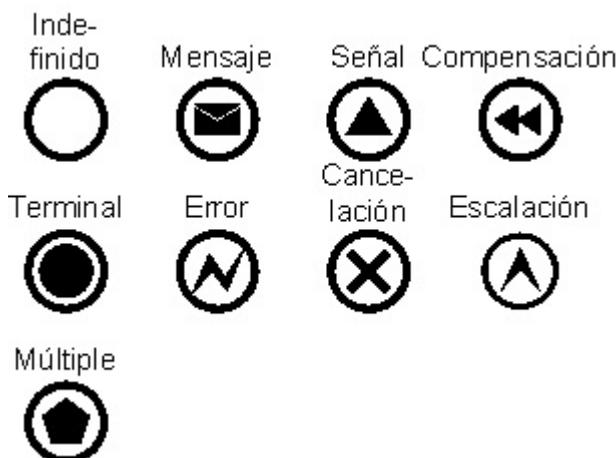


Figura 5.19: Eventos de término clasificados

en normal y anormales

Finalmente nos queda por explicar el comportamiento de un sólo evento de término, el de eventos múltiples. Este evento puede comportarse como cualquiera de las formas descritas más arriba y en diferentes combinaciones de ellas, pero esto no se puede representar en forma gráfica por lo que para la ejecución se tiene que definir en el archivo XML, exactamente su semántica, como lo muestra el siguiente ejemplo:

```
<bpmn:endEvent id=«End» name=»Multiple en event»>
<bpmn:messageEventDefinition id=«Message» messageRef= «messageDef» />
<bpmn:signalEventDefinition id=«Signal» signalRef= «signalDef» /> <!....->
```

5.4.6. Transacciones de negocio vs. técnicas

El concepto de transacciones, errores y compensación lo tratamos extensivamente desde el punto de vista de negocio. En esta sección queremos ver los aspectos técnicos de estas funcionalidades.

Cuando en informática se habla de transacciones, se está pensando en el concepto de ACEP (Atómico - Consistente - Encapsulado - Persistente), es decir, la ejecución de una transacción cumple con todas estas características o no se realiza. La transacción resguarda los datos (encapsulados) en su estado anterior y sólo se realiza la transacción o el conjunto de transacciones si la consistencia de los datos ha sido mantenida. Sin embargo, la actualización de los datos en la base de datos es técnicamente una acción atómica.

En nuestro ejemplo de la reserva de viaje en la figura 5.20 podemos explicar la diferencia entre lo que se entiende por una transacción de negocio versus una técnica. Bajo transacción de negocio se entiende la consistencia de todos los datos de un caso de negocio, es decir una transacción de negocio puede incluir varias transacciones como lo muestra nuestro ejemplo. Al cliente le interesa que todas las reservas sean confirmadas porque una sola no le sirve; confirmación de reserva de vuelo y habitación en el hotel. Sólo si ambas reservas han sido confirmadas puede cargarse a la tarjeta de crédito del cliente el cobro correspondiente. Si una de las reservas falla, deben reversarse todo el resto de las reservas.

Desde el punto de vista técnico el concepto de transacciones se maneja a un nivel inferior (ACEP), en nuestro ejemplo, cada reserva o el caso de la carga de la tarjeta de crédito es una propia transacción, el llamado de servicios puede ser asíncrono, por lo que la consistencia a nivel de proceso de negocio hay que controlarla de otra forma, pero el lector ha de diferenciar claramente entre ambos conceptos. Una vez ejecutada la transacción técnica, no es posible volver atrás, el registro en la base de datos se actualizó definitivamente. Técnicamente no podemos volver atrás, pero si podemos corregir desde el punto de vista de negocio la situación. Por ejemplo, reversar significa hacer una contrapartida, pero no volver atrás técnicamente. Las actividades del tipo compensación se encargan de reversar una situación de negocio, pero técnicamente se anula el caso registrando una contrapartida a la transacción.

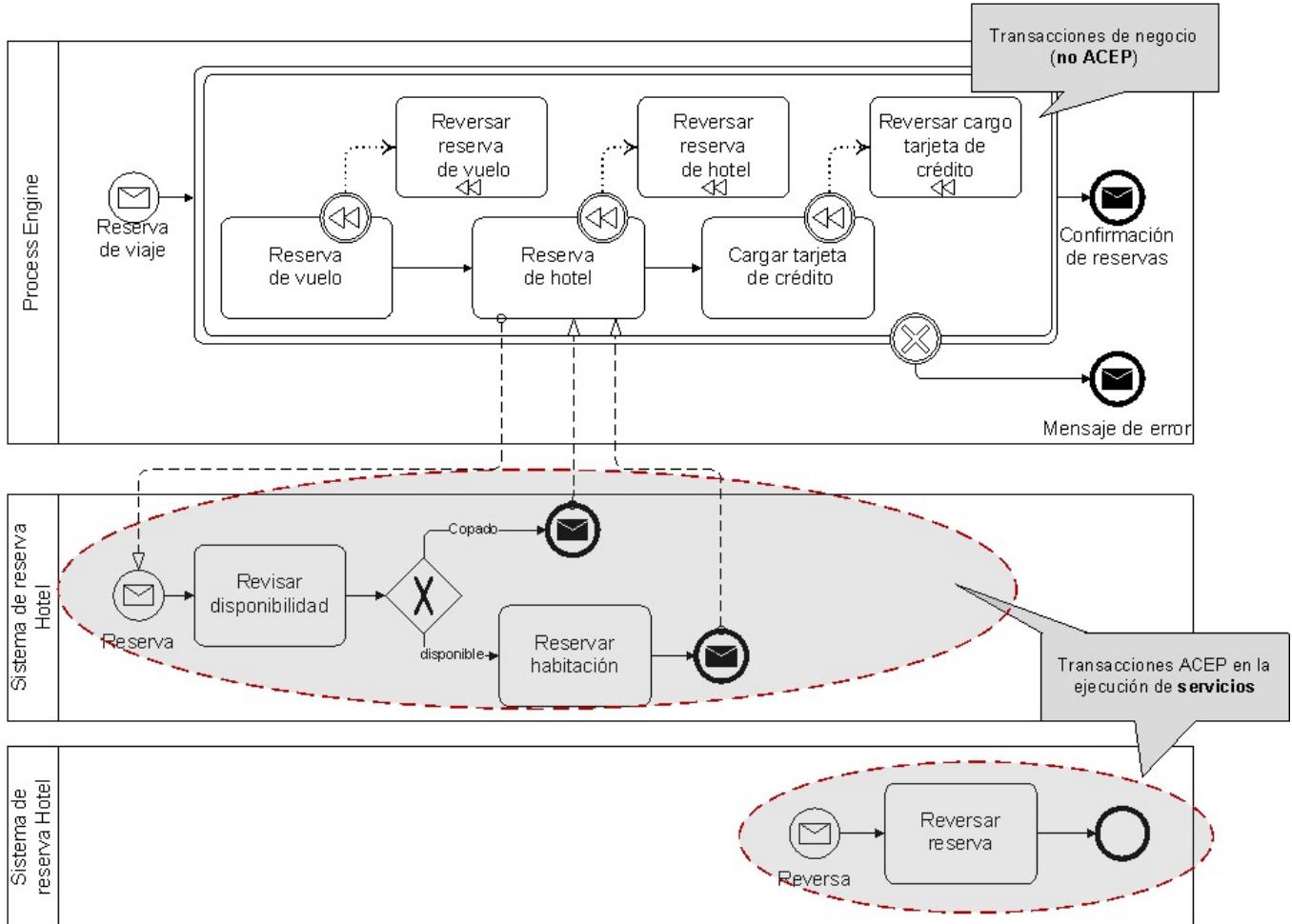


Figura 5.20: Transacciones de negocio y técnicas

Desgraciadamente muchos Process Engines aún no ofrecen servicios de compensación y si pensamos que un servicio de compensación también podría arrojar un error durante su ejecución habría que considerarlo en el modelo técnico. El ejemplo de la figura 5.21 no contempla un tratamiento de excepción en caso que falle la actividad de compensación, es decir el modelo técnico en BPMN habría que ampliarlo si se pretende ejecutarlo directamente.

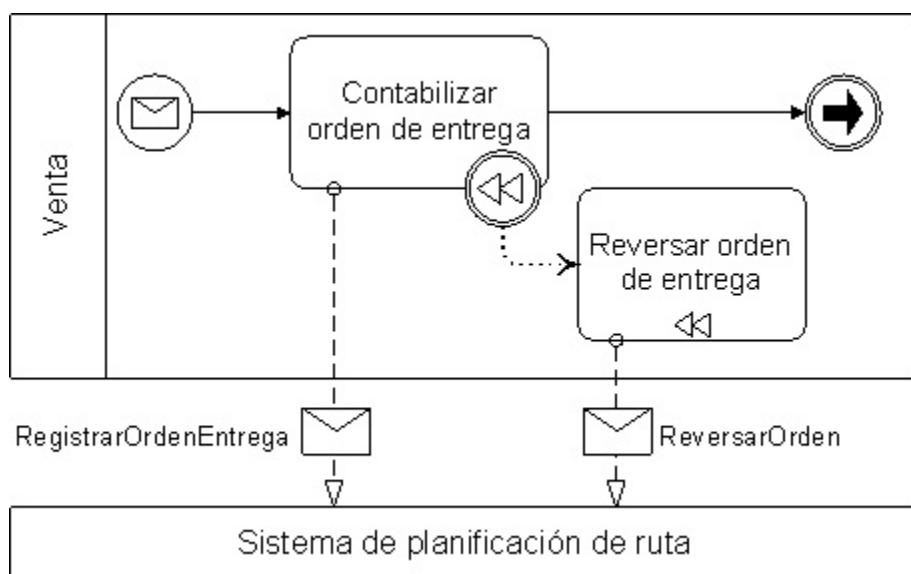


Figura 5.21: Ejemplo de compensación de un llamado de servicio La especificación BPMN 2.0 deja abierto como se maneja el tratamiento técnico del mecanismo de transacción, pero si menciona el concepto de WS-Transaction (en inglés: transacción de webservice). Este estándar define los dos casos que tratamos anteriormente, la transacción de negocio y la transacción técnica:

WS-AtomicTransaction, incluye ACEP transacciones en el mundo de los webservices.

WS-Business-Activity, define el tratamiento de transacciones de negocio con una duración más prolongada que en el caso anterior.

Estos estándares están bien pensados, pero aún no se han desarrollado lo suficiente en la práctica. Como la especificación no norma la tecnología al respecto, habrá que esperar que los proveedores se pongan de acuerdo en un protocolo que permita combinar diferentes tecnologías.

5.4.7. Subprocesos

En la sección 2.8.2 aprendimos que en BPMN se distinguen dos tipos de subprocessos, los incrustados y los reutilizables. Los subprocessos del tipo incrustados tienen una lógica encapsulada dentro de un dominio (en inglés: Scope). Al interior de este dominio, el subprocesso sólo ve sus propiedades y objetos de datos. Estos «Scopes» conforman un área o un paréntesis para el tratamiento de errores, compensación y transacciones. Desde el punto de vista de un Process Engine el subprocesso se encuentra encapsulado en el proceso superior.

En la figura 5.22 ilustramos las vistas que tienen subprocessos en diferentes niveles de abstracción:

El proceso 1 tiene acceso al objeto de datos A, pero no tiene acceso a los objetos de datos B o C.

El proceso 2 tiene acceso al objeto de datos A y B, pero no a C.

El proceso 4 tiene acceso a los objetos de datos A y C.

El evento de error superpuesto en el proceso 2, recibe mensajes de error de los procesos 2 y 3.

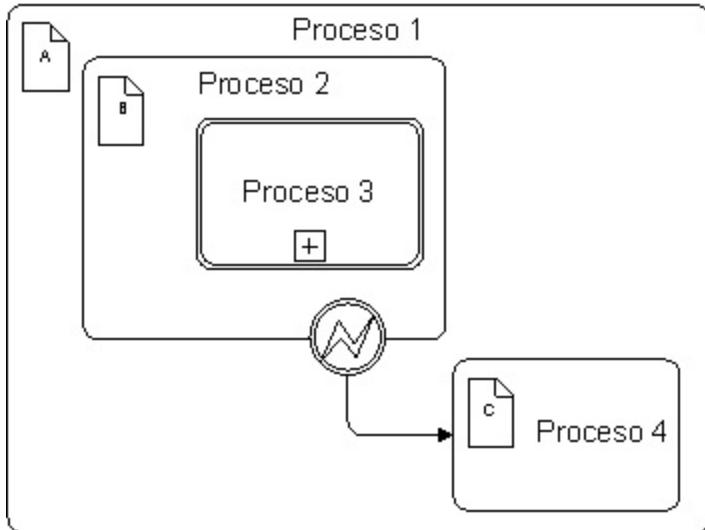


Figura 5.22: Subprocesos, Scope y vista de datos

Los subprocessos reutilizables tienen a diferencia de los subprocessos incrustados sus propias vistas de datos. Si un subprocesso reutilizable requiere datos del proceso superior, estos deben ser traspasados explícitamente, al igual que el llamado de un servicio. En BPMN se declaran estas vistas de datos en «*InputOutputSpecification*».

Queremos hacerle al lector el comentario que a pesar que BPMN es un estándar, los Process Engines pueden comportarse de forma muy diferente con respecto a la forma de como implementan subprocessos. Si una Process Engine trabaja con BPEL, la semántica de ejecución de un subprocesso es siempre vía invocación de Webservices. En realidad BPEL orquesta servicios y no conoce el concepto de subprocessos. El problema que se da en la práctica al convertir todos los subprocessos a Webservices es una pérdida considerable de performance, debido a que cada llamada consume considerables recursos, a pesar que la orquestación no traspasa los límites del Process Engine. Al respecto, hemos conocido casos en algunos de nuestros clientes que han estructurado muy bien sus procesos en BPMN con subprocessos, pero al mapearlos a BPEL técnicamente han resultado un desastre en cuanto a tiempos de respuesta. Es de esperar que los nuevos Process Engines que ejecuten directamente BPMN 2.0 traduzcan el concepto de subprocessos como parte de la especificación y no como servicios que se requieran orquestar.

5.4.8. Loops e instancias múltiples

La forma en cómo la especificación de BPMN 2.0 define loops e instancias múltiples, tiene una cierta semejanza con los lenguajes de programación. En la componente loop se repite la lógica contenida en forma secuencial. El atributo «*test-before*» regula si la «*Expression*» declarada en «*loopCondition*» es revisada antes o después del bucle. La condición «antes» corresponde a un «*while*» en un bucle y la condición «después» a un «*repeat until*» en lenguajes de programación. Si la expresión, en la Expression Language, del Process Engine devuelve el valor «falso»,

termina el loop y el flujo de secuencia abandona la actividad. También se puede declarar una condición que determina un máximo de repeticiones en el loop (loopMaximum).

Las actividades del tipo múltiple son más poderosas que las del tipo loop. La lógica puede ejecutar instancias paralelas en forma simultánea y al mismo tiempo repetir condiciones en forma secuencial como en el caso de actividades de loop. La diferencia a la condición de loop es que el número de instancias tiene que indicarse antes de su ejecución. Esto se puede declarar por una Expression antes de su ejecución o se traspasa una vista de datos llamada «Collection», entonces se crea y ejecuta una instancia por cada objeto de datos declarado en la colección.

Una especialidad en este tipo de actividad múltiple, es que se pueden especificar eventos gatilladores. Normalmente al terminar todas las actividades instanciadas, un Token abandonará la actividad de instancia múltiple. En reemplazo de este mecanismo se pueden utilizar eventos, pero hay que indicar explícitamente cual evento es el que va a ser impulsado. Se puede especificar la condición que impulsará un evento por cada instancia o sólo uno al finalizar la ejecución de todas las instancias. El evento debe ser del tipo «no interrupción». Finalmente se pueden definir construcciones semánticas complejas de múltiples eventos, entonces se evalúan todas la Expressions para determinar el comportamiento de los eventos por impulsarse.

Analicemos ahora el tratamiento de errores o casos de excepción en actividades de instancias múltiples. La figura 5.23 muestra varias alternativas de modelamiento. El problema que tenemos en este caso es con los eventos que interrumpen la actividad en general, porque se puede dar un comportamiento no deseado.

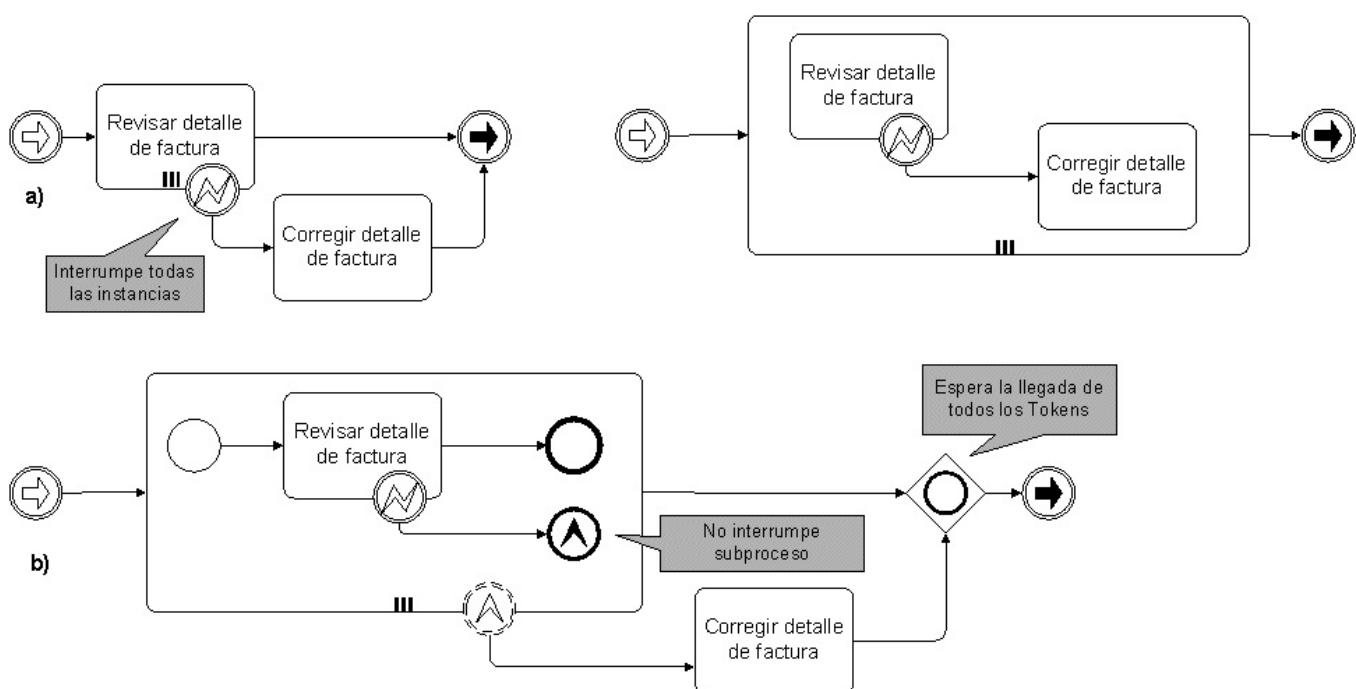


Figura 5.23: Tratamiento de errores en instancias múltiples Supongamos que sucede un error durante la ejecución paralela de instancias, entonces el evento de error va a interrumpir todas las instancias (figura 5.23, caso a)), pero el comportamiento que queremos modelar es que sólo se detenga la actividad con la instancia que contiene el error. Esto lo podemos hacer dentro de la actividad de la instancia en una Expression, pero no es visible en el modelo técnico. También puede darse la situación que el tratamiento de error le corresponde a un usuario ubicado en otro lane, entonces ¿cómo modelarlo? Capturamos el error en la actividad correspondiente y gatillamos un evento del tipo escalación (¡aunque no lo sea!) hacia la actividad que le corresponda. Escogemos para estos fines el evento de escalación del tipo «no interrupción» (figura 5.23, caso b)). De esta forma no interrumpimos el flujo de secuencia del resto de las instancias.

Y ahora ¿cómo controlamos el flujo de los Tokens en este patrón? Casi por magia podemos utilizar el OR-Join porque este espera la llegada de todos los tokens activos, es decir soluciona nuestro problema de sincronización entre instancias múltiples y tratamiento de casos de excepción.

Es posible que el lector encuentre estas construcciones algo complicadas, pero ese es y será siempre el problema de una semántica de ejecución. Si pretendemos que el modelo sea ejecutable con un Process Engine, tenemos que ser exactos en expresar el comportamiento esperado. Técnicamente todas las posibilidades pueden tener sentido, pero desde el punto de vista de negocio no, razón por la cual el ser humano tiene que decir y decidir lo que quiere.

Capítulo 6

Técnicas de Análisis y Mejora

6.1. Introducción

Con el presente capítulo queremos entregar al lector instrumentos que puede utilizar para analizar los procesos en el momento que se requiera hacer un nuevo diseño con el objetivo de cambiar la situación actual a una mejor.

Bajo técnicas de análisis y mejora entendemos todas aquellas técnicas que se emplean para analizar y mejorar los procesos organizacionales desde las perspectivas: Tiempo-Calidad-Costos. Lógicamente que detrás de cada una de estas dimensiones hay todo un mundo de definiciones, técnicas e indicadores, las cuales no podemos tratar en profundidad en este trabajo. Nuestra intención es abordar aquellos aspectos con los cuales nos enfrentamos al levantar, modelar y evaluar los procesos existentes. Para poder abordar estas técnicas, es importante que el analista de procesos no se confunda, con disciplinas especializadas en aplicar metodologías de mejora continua como lo son Six-Sigma o Kaizen, pero también distinguir enfoques radicales como lo es la reingeniería. Una reingeniería poco tiene que ver con el modelamiento de procesos como lo hemos tratado hasta el momento (ver sección 6.2). Nuestro objetivo es tratar aquellas técnicas de análisis y mejora que nos apoyan en el rediseño de los procesos a partir de una situación actual documentada y validada. Todas estas propuestas de mejora pueden contribuir en mejorar los tiempos de ciclo, la calidad de los servicios y en algunos casos a bajar los costos de las actividades de un proceso, pero si queremos introducir una metodología de mejora continua y olvidarnos de la situación actual o rehacer un proceso nuevo, no

237 es dominio ni materia de un proyecto de modelamiento de procesos como lo son proyectos de BPMN u otra notación.

6.2. Reingeniería, Rediseño y Mejora

El término «Reingeniería de Procesos» (en inglés: Business Process Reengineering, BPR) fue propagado por una publicación de Hammer&Champy en el 93[HamCham93], que se hizo famoso como respuesta al hecho de cómo las empresas tradicionales podían enfrentar la prolongada recesión que se venía arrastrando desde mediados de los años 80 en los países occidentales, amenazados por la eficiente competencia asiática. Al respecto Hammer&Champy dejan claro cuál es el «driver, o el Leitmotiv» para aplicar una reingeniería[HamCham93]:

«Reingeniería de Procesos significa quebrar con paradigmas antiguos,

procedimientos obsoletos y orientarse fundamentalmente hacia la creación de valor para el cliente, al pensar en reestructurar la nueva forma de organizar el trabajo.»

El método de la Reingeniería de Procesos (BPR) fue descrito por Hammer&Champy como la “reconsideración fundamental y la reorganización radical” para lograr una mejoría drástica en el desempeño, los costos y los servicios. Hammer&Champy recomiendan que se deben observar los procesos completos de una organización, desde la adquisición, pasando por la producción, la venta y la distribución. La empresa debe concebirse y reconstruirse como un conjunto de procesos.

Los principales aspectos de la reingeniería de procesos son:

Orientación a la satisfacción del cliente (Tiempos de respuesta, calidad de productos y servicios, costos).

Reconsideración fundamental de la organización del trabajo (actividades, flujos, responsabilidades).

Considerar las capacidades de TI para mejorar la eficiencia de los procesos.

El BPR requiere de grandes esfuerzos, consume muchos recursos y demanda una gran coordinación de participantes de todos los niveles en una organización, razón por la cual se clasifica en una de las categorías de proyectos de alto riesgo. El BPR debería de restringirse a aquellos procesos de negocio que tienen una importancia estratégica y que su desempeño actual es altamente deficiente. Pero, ¿qué podríamos entender por deficiente y cuando se justifica una reingeniería de procesos?:

La competencia supera sin problemas a la compañía

6.2. REINGENIERÍA, REDISEÑO Y MEJORA 239

Los productos y servicios no satisfacen las necesidades de los clientes:

- Se requiere de una nueva solución orientada a la demanda del mercado (clientes)
- Una nueva solución requiere el diseño de un nuevo proceso

Existen muchos conflictos dentro de la organización:

- Muchas reuniones sin resultados
- Poder informal es más alto que el formal
- Comunicación excesiva fuera de la línea

Un BPR sólo se puede llevar a cabo a través de un proyecto con el respaldo del directorio de la empresa. Ha de considerarse que un BPR impacta en forma cultural, procedural y estructural, por lo que se requiere incluir otras disciplinas al proyecto como «Gestión del Cambio Cultural», «Gestión del Conocimiento», «Programa de Capacitación», etc..

Generalmente los procesos en las organizaciones no se encuentran «tan mal» organizados, y por lo que en muy contadas ocasiones podría justificarse un proyecto de reingeniería de procesos. Por otro lado el BPR no considera cambios graduales o

incrementales, por lo que ha ido perdiendo importancia como técnica de reorganización y mejora en los últimos diez años.

Muchas veces se confunden los conceptos de «Reingeniería» y «Rediseño», se emplean como sinónimos pero no lo son. El rediseño de procesos, no es tan radical como la reingeniería, puede por ejemplo aplicarse a una parte del proceso de negocio y tiene como objetivo mejorar el grado de competitividad a través de técnicas de optimización de procesos. El mayor impacto de un rediseño se tiene si el análisis comienza con los eventos generados por los clientes y los resultados que llegan a ellos, por ejemplo solicitudes, pedidos, pagos, reclamos, etc.. Las dimensiones de optimización en el rediseño son: reducción de los tiempos de ciclo, mejoramiento de la calidad de los productos y servicios, reducción de costos.

El rediseño establece los cambios que deberán efectuarse en la situación actual y detalla cómo se ejecutarán los nuevos procesos. Es la fase más importante, ya que se definirán las nuevas formas de operar y su desempeño. ¿En qué ámbitos influye el rediseño?:

Estructural: cambio en el proceso mismo (cambian las operaciones, se eliminan duplicidades, etc.)

Productividad: Análisis de ciclo y costeo de actividades.

Responsabilidades: se modifica la asignación de responsabilidad (personal, centralizar o descentralizar responsabilidades, etc.)

Integración: mejorar el grado de integración entre la capa de la estrategia, operacional (procesos) y tecnología (producción y TI)

Incorporación de tecnología: automatización de procesos, aplicación de tecnologías móviles, integración de sistemas, etc..

Finalmente bajo el término de «Mejora» se entiende en forma abreviada en BPM, el BPM-Governance o dicho en español «El círculo virtuoso de mejora continua por medio de gestión por procesos». El concepto de la «Mejora Continua» está inserto dentro de la gestión diaria de operaciones y a diferencia de la técnica de rediseño no requiere de la formulación de un proyecto. El ciclo de la implementación de la mejora queda en manos de los responsables del negocio y no consumen recursos adicionales a los propios. Algunos de estos conceptos de mejora continua se conocen bajo los nombres de Six Sigma, Kaizen y Total Cycle Time, pero también podemos sumar a estas técnicas el sólo monitorear el rendimiento de los procesos a través de indicadores de ciclo u otros e iniciar iniciativas de mejora cuando se detectan desviaciones al comportamiento esperado (BPM-Governance). El concepto de mejora continua está limitada a cambios pequeños como reglas de negocio, procedimientos locales, redistribución del volumen de trabajo, simplificación de formularios, etc.. Si los cambios propuestos por la «Mejora Continua» impactan

sobre la estructura de los procesos, traspasan los límites de responsabilidad del área, impactan sobre la tecnología, o bien requieren de recursos adicionales, la propuesta de mejora pasa a un proyecto de «Rediseño».

Describir y analizar todos estos conceptos de mejora continua sobrepasaría los objetivos de un libro de notación y modelo de procesos, como lo es en este caso con BPMN. El lector puede consultar la literatura de BPM o especializada al respecto, si le interesa seguir profundizando esta materia.

Sin embargo, en un proyecto de modelamiento de procesos sea con BPMN u otra notación, siempre se espera o se ordena analizar el contexto en vías a mejorar el rendimiento de los procesos actuales cuando se implementen. No es suficiente que el analista aprenda la notación en que va a modelar los procesos, sino que también para poder proponer un nuevo diseño del proceso, debe saber qué analizar y como medir para después poder evaluar. A estos instrumentos le llamamos en este capítulo «Técnicas de Análisis y Mejora» y serán materia de la siguiente sección.

La figura 6.1 muestra una tabla con las principales características que diferencian los tres enfoques principales de mejora de procesos:

Característica	Reingeniería	Rediseño	Mejora
Enfoque	Proceso nuevo	Reestructuración	Mejora evolutiva
Punto de partida	Proceso existente	Proceso existente	Proceso existente
Objetivo del cambio	Cambio radical, satisf. Cliente	Rediseño de una parte del proceso	Actualización, Eficiencia o satisf. cliente
Tipo de cambio	Radical	Estructural	Incremental
Periodicidad del cambio	Descontinuado	Intervalos intermedios	Continuo
Organización del cambio	Proyecto	Proyecto o grupo de trabajo	Dentro de operaciones
Impulsor del cambio	Directorio	Dueño de proceso	Cualquier actor
Impacto del cambio	Transversal	Proceso, subprocesso	Dentro de un subprocesso
	Cultural	Cultural	Cognitivo
	Procesal	Procesal	Procedimiento, Regla de negocio
	Estructural	Estructural	Costo, calidad, tiempo
Riesgo	Alto	Medio	Bajo

Figura 6.1:
Reingeniería - Rediseño - Mejora Continua

6.3. Clasificación y Tipos de Mejora

6.3.1. Análisis de estructura

Con el análisis de estructura se busca mejorar el desempeño de los procesos sobre todo con miras a reducir los tiempos de ciclo y mejorar la calidad de los servicios de los procesos. Para estos efectos podemos revisar:

El orden de las actividades en un proceso

Si existen redundancias

Actividades, procedimientos o reglas de negocio obsoletas

Flujos complejos que se pueden simplificar

Un estudio de Bleicher[Gad10] nos muestra en las figuras 6.2 y 6.3 las posibilidades que tenemos para reestructurar los procesos:

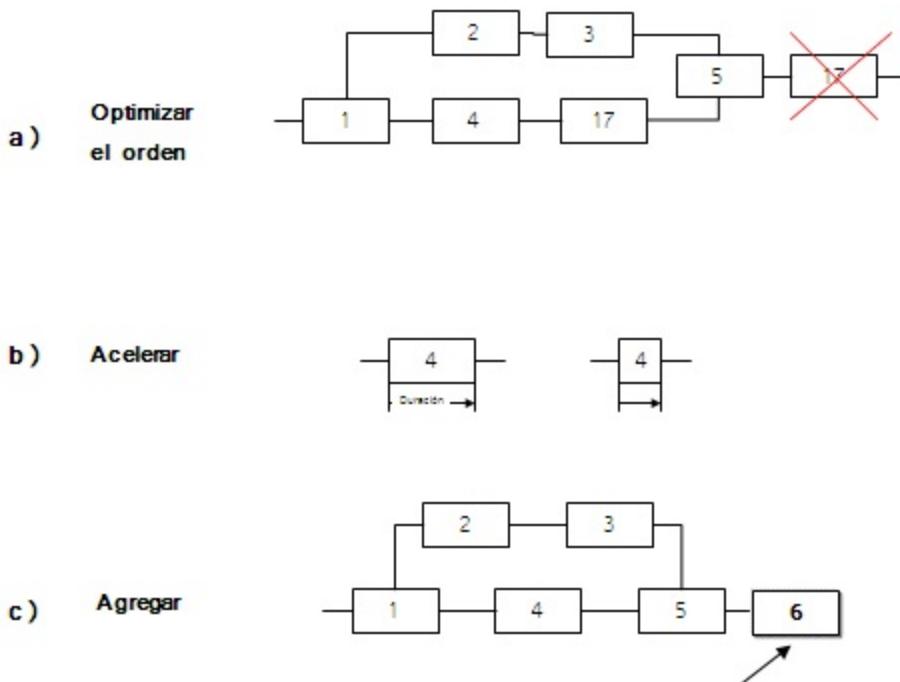


Figura 6.2: Análisis de

estructura según Bleicher (parte 1)

En el flujo a) podemos revisar si las actividades se pueden iniciar antes. En el ejemplo se muestra que la actividad (17) puede realizarse después de la (4). En este ejemplo el tiempo de ciclo del proceso podría reducirse al ejecutar la actividad (17) antes de la (5).

En el caso b) podemos dotar de mayores recursos la actividad (4), con lo que logramos agilizar el tiempo de ejecución de esta actividad. Este caso representa el típico «cuello de botella», cuando un usuario tiene mucho volumen de trabajo y otras tareas tienen que esperar a la finalización de ésta.

El flujo c) muestra una posibilidad bastante poco considerada en la práctica, porque agregar una actividad aumenta el costo de los recursos, pero puede mejorar notablemente la calidad del servicio y con esto el grado de satisfacción de cliente.

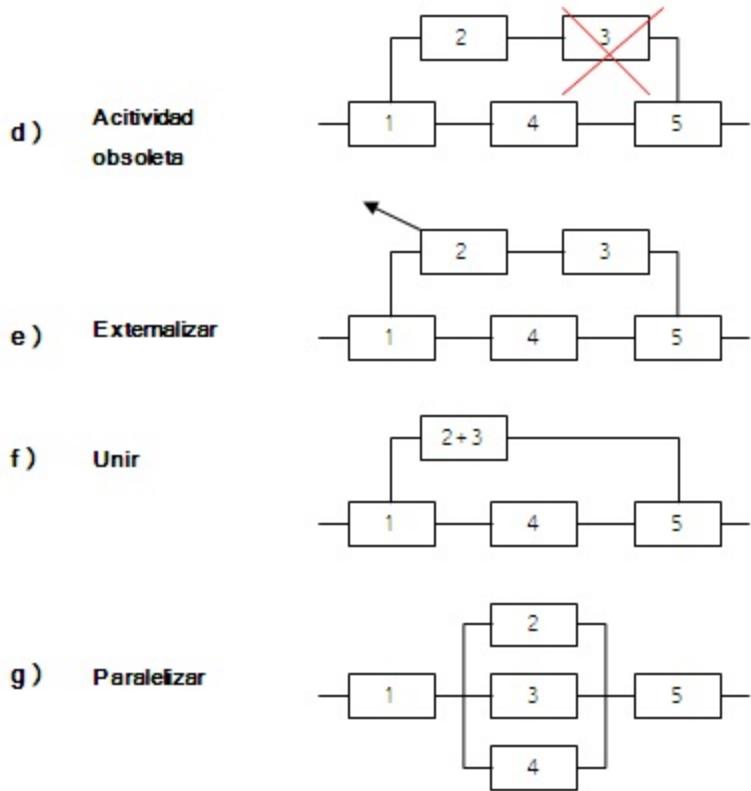


Figura 6.3: Análisis de estructura según Bleicher (parte 2)

El flujo d) muestra como se acorta el ciclo si podemos desistir de una actividad en el proceso. Para revisar si encontramos actividades obsoletas tenemos que preguntar en las reuniones de análisis: ¿Qué pasaría si desistimos de esta actividad?

El caso e) muestra la posibilidad de externalizar un servicio si su realización es más eficiente entregarlo a especialistas. Piense en el caso de la necesidad de elaborar o revisar contratos de negocio, contratos de empleo, finiquitos etc.. Si el volumen de una actividad es pequeño, pero se requiere de mucho conocimiento específico para resolverla, es un candidato a externalización. En nuestro caso sería contratar los servicios a un estudio jurídico.

El caso f) muestra la posibilidad de unir actividades. Supongamos que la entrada de una factura pasa por dos revisiones formales, revisión de integridad de datos y existencia de una orden de compra (2 y 3) antes que sea enviada al ejecutivo de área. Si ponemos a disposición la información necesaria para que se puedan revisar en conjunto, nos ahorraremos el traspaso de una tarea a otra.

Finalmente el caso g) muestra la posibilidad de paralelización de actividades en un flujo de procesos. Si logramos paralelizar actividades podemos reducir el tiempo de ciclo de un proceso.

6.3.2. Análisis de tiempo de ciclo

Como hemos visto, el concepto de tiempo de ciclo dice relación con el tiempo que toma el proceso en ejecutar una instancia, desde su inicio hasta el fin del proceso. Si observamos con detenimiento la figura 6.4, la interpretación del diagrama supone que la actividad “Revisar resultado” se ejecuta inmediatamente de concluida la actividad “Producir resultado” (o “Corregir resultado” según sea el caso), cuestión que en el mundo real pocas veces ocurre ya que usualmente el “resultado” producido (o corregido) quedará esperando para ser revisado.

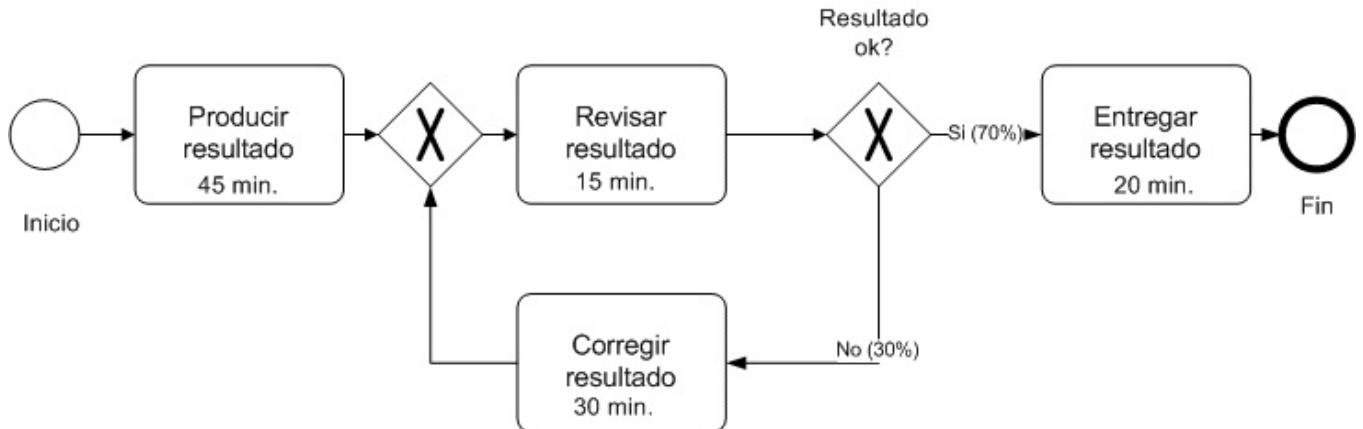


Figura 6.4: Análisis de tiempo de ciclo

De no haber esa espera para que el resultado sea revisado, el tiempo de ciclo será la suma -probabilidades e iteraciones de por medio- de tiempos asociados a las actividades ejecutadas.

En caso contrario, es decir frente a la existencia de esperas o detenciones en el flujo, el tiempo que permanezca la instancia en espera para ser atendida por la actividad siguiente aumentará el tiempo de ciclo. En general se considera que las esperas o detenciones en el flujo no agregan valor para el cliente, toda vez que no se está efectuando trabajo efectivo sobre la instancia. ¿Existe algún cliente que guste de pagar por estar esperando?

A la suma de tiempos de las actividades que agregan valor se conoce como tiempo de valor agregado. Normalmente se expresa como la fracción o porcentaje respecto del tiempo total o tiempo de ciclo.

Así, muchas veces, para obtener mejoras en el tiempo de ciclo se pone más atención en las esperas o detenciones que afectan a la instancia, que en las actividades mismas del proceso.

Hay que considerar que los tiempos de duración de las actividades son variables aleatorias, usualmente con distribución de probabilidades exponencial, por lo que cuando se indica el tiempo de la actividad en realidad se está haciendo referencia a un tiempo promedio observado de duración de la actividad (estimación del tiempo esperado de duración de la actividad). También hay que tener presente que la

variabilidad se ve disminuida en actividades automatizadas. En el mercado existen herramientas específicas de simulación de procesos, muy populares en el ámbito de la manufactura, que permiten obtener respuestas a preguntas como: ¿Cuál es la probabilidad que la instancia sea procesada antes de X minutos?, ¿Qué pasa si se aumentan los recursos (servidores) que ejecutan una actividad?, etc.

Las colas o esperas, también conocidas como buffers o amortiguadores, existen en los procesos normalmente debido a:

El diseño del proceso consideró la existencia de la espera o almacenamiento de instancias, para ser ejecutadas por la actividad siguiente. El proceso responde con una cola ante la incapacidad de éste para procesar el flujo al que está siendo sometido. Esto podría ser también parte del diseño.

Como ejemplo del primer caso, tenemos el proceso de embarque de pasajeros en un vuelo comercial, en que los pasajeros esperan para ser llamados a subir al avión. Por su parte el proceso de atención en la caja de un banco constituye un ejemplo donde el proceso responderá con una cola cuando el flujo (demanda de clientes para ser atendidos) superó la capacidad de atención de los cajeros.

Conceptos asociados a estos son el “Cuello de botella” que dice relación con una reducción en la capacidad de flujo del proceso, y el “Bloqueo” que se refiere a una detención de las actividades ya que una de ellas no tiene “donde dejar” la instancia ya procesada.

En BPMN, para el analista del negocio resultará interesante además de consignar en un token el tiempo de valor agregado, consignar también el tiempo de ciclo o los tiempos de espera de las instancias, entendidos éstos como aquellos tiempos usados durante la ejecución del proceso pero no se agrega valor para el cliente.

La existencia de colas conlleva la definición de reglas para resolver cuál de las instancias que esperan será la próxima en ser atendida por la actividad siguiente, como ser: atender al más antiguo primero (FIFO:First In First Out), los clientes Premium primero, los ancianos y niños primero, pasajeros con asiento entre tal y cual fila primero, etc.

En términos generales, además de reducir el tiempo de ejecución de las actividades, las recomendaciones a fin de reducir el tiempo de ciclo en un proceso dicen relación con:

Reducir las interrupciones del proceso, entre ellas el tiempo de preparación (seteo) de máquinas. Cada vez que se interrumpe el proceso quedan instancias esperando ser procesadas, una vez restablecido el flujo el proceso tomará un tiempo en recuperar su ritmo anterior a la interrupción, sin duda en ambos casos el tiempo de ciclo de las instancias que se encuentren en proceso se verá afectado negativamente. Piense en una planta de producción, normalmente éstas están diseñadas para un óptimo

desempeño mientras se encuentran en funcionamiento. En caso de una detención del flujo del proceso podría tomar horas o días en retomar el funcionamiento normal.

Eliminar los cuellos de botella. Un cuello de botella se produce en una actividad, disminuyendo el flujo y por lo tanto generando una cola o espera previa a la actividad. Esto ocurre porque el proceso no se encuentra bien balanceado o por una baja capacidad de la actividad para atender la demanda o flujo. Balancear adecuadamente el proceso, aumentar la capacidad de proceso de la actividad, duplicar en paralelo la actividad, serían recomendaciones a considerar frente a un cuello de botella.

Eliminar las colas o almacenamientos. Como vimos las colas o lista de espera se encuentran asociados a los cuellos de botellas, la cola se produce naturalmente donde el flujo se hace más lento y así un aumento en el tiempo de ciclo del proceso, por ello es que si busca disminuir el tiempo de ciclo ponga atención a las colas y elimínelas cuando pueda. En algunos procesos la cola es parte del diseño, por ejemplo en una empacadora al final del proceso se apilarán las unidades hasta completar la cantidad suficiente para completar una unidad de empaque mayor.

Cambiar el orden de las actividades. Pensemos en el proceso de atención de un paciente en una unidad de emergencia, donde el médico indica realizar a un paciente una radiografía, para lo cual el paciente debe ser llevado a la unidad de imaginología, y una toma de muestra de sangre para ser procesada en el laboratorio. Para ambas actividades requerimos del paciente, por lo que –supongamos que- no las podemos ejecutar en paralelo. Si llevamos primero al paciente a imaginología y luego se extrae la muestra de sangre, el laboratorio comenzará a efectuar el análisis de ella más tarde que si primero se toma la muestra y mientras es enviada al laboratorio llevamos al paciente a tomar la radiografía. Este cambio en la secuencia de las actividades claramente puede llevar a un menor tiempo de ciclo del proceso de atención del paciente. Diseñar actividades en paralelo. Acá el principio es la simultaneidad, es decir que en una misma unidad de tiempo se puedan estar ejecutando más de una tarea a la vez sobre una misma instancia, tal como se explica en la letra g) de la Figura 6.3, considerando un AND-Split y un AND-Join.

Juntar dos o más actividades en una. Como se explica en la letra f) de la Figura 6.3, en algunos casos se recomienda la unión de dos o más actividades en una, a modo de ejemplo que el control de calidad de una actividad sea parte de la misma actividad, esto conlleva entregar responsabilidad al operador en la calidad del trabajo que ejecuta.

Otra recomendación es analizar el tiempo de ciclo de un proceso con métodos de Gestión de Proyectos, viendo el proceso como una red, identificando la ruta crítica y luego privilegiando la reducción de tiempos de las actividades o tareas de la ruta

crítica (CPM: Critical Path Method), esto aplicado a las distintas instancias identificadas que serán atendidas por el proceso.

Observe que una reducción del tiempo de ejecución de una actividad no necesariamente aportará una reducción en el tiempo de ciclo del proceso. Esto sólo ocurrirá si la actividad se encuentra – para alguna instancia- en una ruta crítica del proceso.

Por último tenga presente que el analista de procesos, siempre deberá estar observando las tres dimensiones ya mencionadas de desempeño de los procesos: tiempo, calidad y costo. Por ejemplo una reducción inadecuada del tiempo de ciclo puede llevar a una pérdida de calidad y con ello un aumento de los costos, un aumento del tiempo de ciclo por incorporación de una actividad de inspección temprana, puede llevar a un aumento de calidad y también una reducción de costos, etc.

6.3.3. Análisis de costeo de actividades

Las empresas o instituciones proveen sus productos y/o servicios a sus clientes a través de procesos de negocio.

Los procesos durante su ejecución consumen actividades y las actividades consumen recursos.

La idea central de costeo por actividades es asignar el consumo de recursos a cada actividad (La obtención de esta información es un proceso separado al de modelamiento), por ejemplo: consumo de materiales, obra de mano, energía, tiempo de máquina, etc., obteniendo así un costo para cada actividad. La unidad encargada de monitorear los costos debería observar las desviaciones entre el costo esperado y el costo observado en la ejecución de las actividades.

En cada proceso se contabiliza la cantidad de actividades que se requieren para su ejecución. Y entonces, en términos sencillos, si tenemos el costo de cada actividad podremos obtener el costo de producto o servicio producido por el proceso (costo de la instancia).

Con BPMN podemos asignar el costo promedio de cada actividad, a través de la inclusión de un atributo. Si las instancias van acumulando los costos al término de un período, se puede conocer el costo de la producción en el proceso. Estos cálculos se encuentran sobre la observación de un proceso en particular, y se puede incluir en un Process Engine como indicador en un cuadro de mando (BAM), como se hace para tiempos de ciclo en forma agregada.

Un objetivo primario de toda empresa u organización es gestionar sus costos.

Veamos a continuación, desde la perspectiva de BPM, algunas recomendaciones:

- Eliminar del proceso aquellas características del producto o servicio que no agregan valor para el cliente.

Estas características innecesarias consumirán recursos que no serán valorados por el

cliente. Incluso el producto o servicio podría contener una característica que destruye valor. ¿Cuántas veces ha reflexionado que determinado producto sería mejor que no tuviera tal característica?

- Aumentar el uso de los recursos

Las mejoras en el ámbito de disminuir el tiempo de ciclo del proceso, entre otras, apuntan a aumentar el uso de los recursos, es decir a aumentar la productividad de la capacidad instalada. Al disminuir el tiempo de ciclo se tiene la oportunidad que con los mismos recursos fijos (instalaciones, máquinas, personal a contrato fijo, etc) se puede obtener una mayor producción de unidades, disminuyendo así el costo unitario de esas unidades.

- Aumentar la calidad en todo el proceso

En este punto el criterio es detectar la falla apenas esta se produzca, de modo de no seguir consumiendo recursos mientras no se repare la falla o se deseche. En definitiva: ¡Hacerlo bien, y a la primera! Esto disminuirá los costos por reprocesos, reparaciones, garantías posventa, des prestigio de la marca, pérdida de clientes, etc. Si bien la mejora en la calidad del proceso podría aumentar inicialmente los costos, se acepta que asumiendo una curva de aprendizaje normal, al cabo de un tiempo generará beneficios que también se manifestarán en reducción de costos del proceso. El lector podrá encontrar en otras disciplinas, como Lean Manufacturing, el desarrollo de técnicas que podrá aplicar a la mejora de costos de los procesos.

6.3.4. Análisis de responsabilidades

El análisis de responsabilidades estudia la relación entre las actividades del proceso y su respectiva asignación de responsabilidades (unidades organizacionales, roles, cargos). En organizaciones grandes y antiguas, como también en la administración pública nos encontramos con estructuras jerárquicas y burocráticas que se pueden reducir, liberando de esta forma actividades que retienen el proceso y no cumplen otra función que confirmar o aprobar un documento que elaboró un ejecutivo o usuario de negocio. La mejor forma de ver los potenciales de optimización que existen al respecto, es el estudio de un ejemplo. Para estos efectos vamos a reproducir un ejemplo ilustrativo que presentó Gadatsch en su libro de BPM[Gad10]. El caso de estudio trata del proceso de «Venta de materiales de una fábrica de construcción de maquinarias». Observando la figura 6.5 el lector podrá seguir la numeración de cada uno de los pasos del proceso:

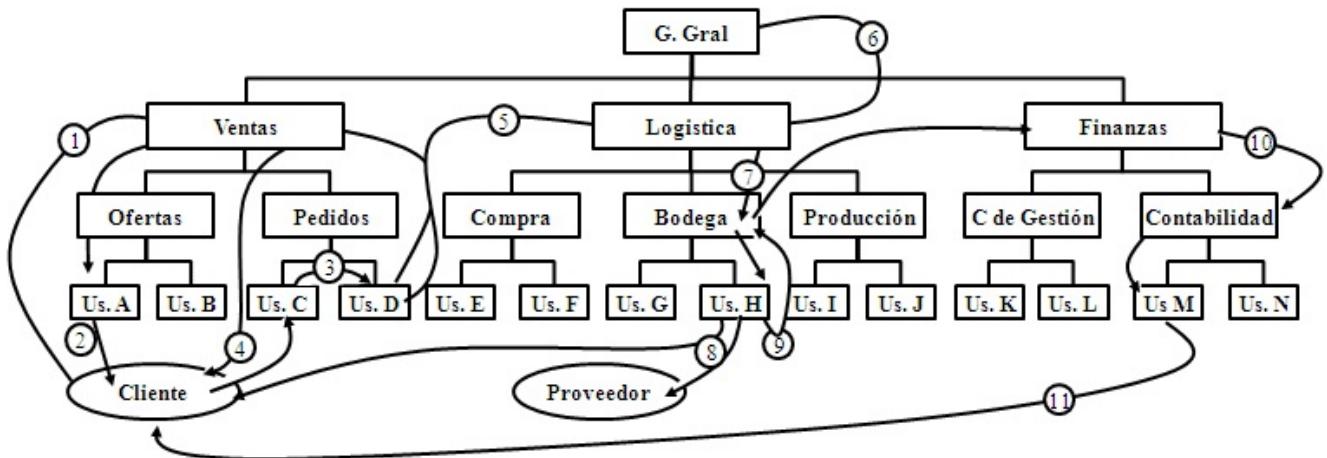


Figura 6.5: Análisis de responsabilidades: Antes de la optimización

1. El proceso comienza con el gerente de ventas quién recibe primero las solicitudes de los clientes.
2. Luego se envía una oferta al cliente, pero a través del usuario A. Antes de enviar la oferta al cliente el encargado de cuentas (jefe de ofertas) debe dar el visto bueno. Como el encargado de cuentas se encuentra a menudo de viaje, la oferta queda retenida hasta que vuelve a su lugar de trabajo.
3. Si el cliente emite una orden de compra (OC), la revisa primero el usuario C en forma manual y luego se la envía al usuario D para que la ingrese en el sistema de administración de órdenes de compra.
4. El cliente recibe una confirmación de su OC, después que el gerente de ventas haya revisado y aprobado la oferta.
5. Luego de haberse registrado la OC en el sistema, llega la OC al gerente de logística. El Gerente decide personalmente si el material lo pide a bodega, si se debe confeccionar por terceros, o si se debe construir en la planta.
6. Si tiene dudas solicita una reunión a la gerencia general para exponer el caso.
7. En este caso el jefe de bodega recibe la orden de entrega de material, pero como se encuentra con licencia médica, libera la entrega del pedido al próximo día y se la envía al usuario H.
8. El usuario H prepara el despacho del producto al cliente y al mismo tiempo genera el pedido de reposición del producto a un proveedor. 9. Luego del despacho el usuario H envía a Bodega el comprobante de despacho y éste a finanzas.
10. El gerente de finanzas entrega el comprobante al jefe de contabilidad y el jefe de contabilidad al contador. Como el gerente de finanzas está sobre cargado de

reuniones con la gerencia general, a veces tardan días para que despache su trabajo al contador.

11. El usuario M emite la factura y se la envía al cliente.

Este proceso consta de 11 pasos y ahora nos podemos preguntar si existe potencial de mejora. Las debilidades del proceso son fáciles de identificar, principalmente en la burocracia de aprobar pasos operativos por altos funcionarios de la empresa:

Personal de la alta dirección toma decisiones operativas.

Muchos usuarios involucrados en el proceso.

Poca comunicación entre los ejecutivos del negocio.

En ausencia de personal, no hay una reglamentación de reemplazo.

La evaluación del análisis nos da pautas claras sobre las mejoras que se pueden introducir: Principalmente en el cambio de asignación de responsabilidades:

La gerencia general no debería estar involucrada en el flujo operacional. Ejecutivos de mandos medios, sólo deberían intervenir en casos excepcionales. El cliente tiene sólo contacto directo con su ejecutivo de cuentas.

Los usuarios de las diferentes áreas colaboran, para casos normales, sin intervención de sus ejecutivos.

Los usuarios se encargan de procesar completamente sus actividades. Si aplicamos estos principios de gestión a nuestro proceso en estudio, se daría un flujo de proceso como lo muestra la figura 6.6:

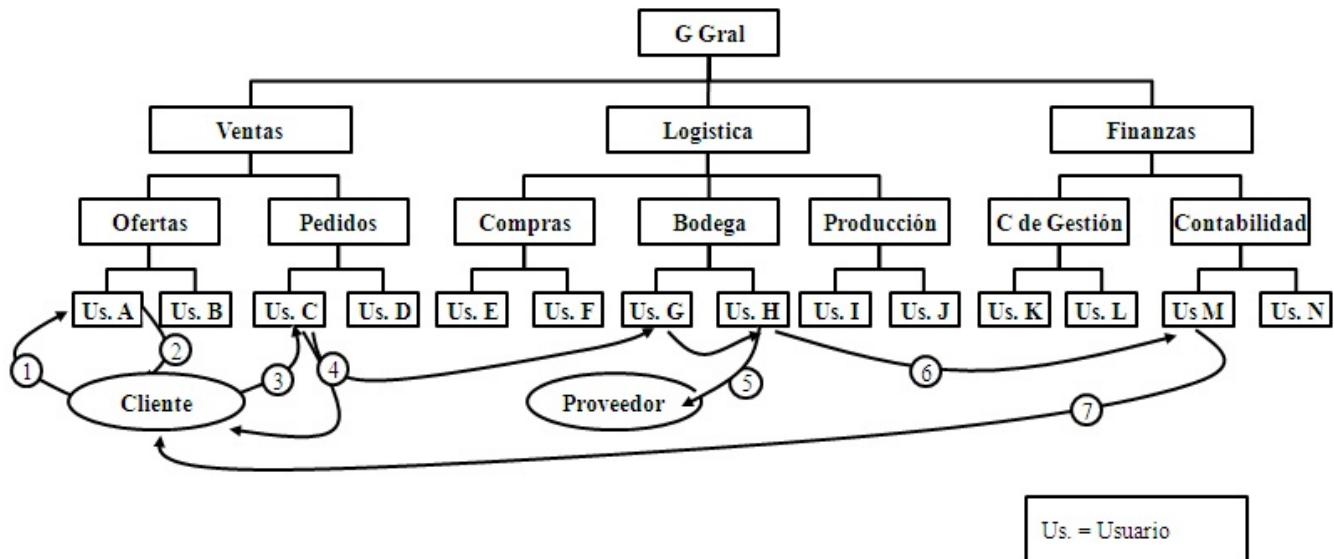


Figura 6.6: Análisis de responsabilidades: Después de la optimización

El proceso rediseñado lo podemos acortar en el presente caso de 11 a 7 pasos y quedaría optimizado como enumerado a continuación:

1. El proceso comienza en que el usuario del área de ventas recibe la solicitud del cliente y elabora completamente una oferta al cliente.

2. El mismo usuario (A) envía la oferta al cliente.
3. Si el cliente acepta la oferta y envía una OC a la empresa la recibe y procesa el usuario C, quién la registra en el sistema de administración de órdenes de compra.
4. El usuario C se comunica con las áreas de compra, producción y bodega con el objetivo de coordinar si es necesario producir o si el producto se encuentra en bodega para calcular una fecha de entrega. Luego el mismo usuario C le envía una confirmación de la OC junto a una fecha de despacho al cliente.
5. En este caso el producto se despacha de bodega. Como en este día el usuario C no se encuentra en su oficina, lo reemplaza el usuario H, quién se encarga del despacho y de generar una orden de reposición del producto para la bodega. 6. El usuario H informa al usuario M en contabilidad.
7. El usuario M elabora la factura con la información entregada y se la envía al cliente.

Si los procesos de negocio son intensivos en la toma de decisiones, la técnica de análisis de responsabilidades constituye un instrumento importante para revisar los potenciales de optimización en el flujo de la toma de decisiones. En muchas ocasiones los altos niveles de dirección se involucran cuando los volúmenes del negocio sobrepasan los casos normales. La automatización de los procesos por ejemplo para ejecutivos en dispositivos móviles (tablets) pueden en estos casos agilizar bastante los procesos de decisión, porque estos pueden acceder a la información independiente del lugar dónde se encuentren.

Capítulo 7

Recomendaciones para Empezar

7.1. Desarrolle su Estilo

Nuestra intención fue de explicarle la notación de BPMN en detalle y de orientar al lector con apoyo de nuestro BPMN Framework como modelar en los tres niveles de abstracción indicándole los principios de modelamiento y los conceptos que hay detrás de la especificación de este estándar. Más allá de todo esto desarrollamos una metodología de como aplicar BPMN en la práctica y le mostramos con el capítulo 6 cuales técnicas de análisis puede emplear en un proyecto de rediseño de procesos. ¡Ahora le toca a usted! Debe tener claro cual será el objetivo de aplicar BPMN y de acuerdo a este objetivo definir sus propias convenciones. Por supuesto que usted puede recurrir a nuestro Framework y a todas nuestras recomendaciones, pero aun así, existe un amplio margen que debe acotar de acuerdo a su realidad organizacional. El estándar no es una guía que pueda aplicar sin antes pensar, planificar y definir, por ejemplo de decidir qué tipo de objetos va a utilizar para qué nivel y en qué situaciones.

Su estilo, no lo podrá desarrollar en forma abstracta, sino aplicándolo concretamente en procesos reales de su organización. Le recomendamos al principio comenzar con procesos simples, por ejemplo:

- Solicitud de vacaciones
- Recepción y pago de facturas
- Compra de insumos de oficina

253 Algunos proyectos de BPMN comienzan de inmediato levantando y documentando los procesos críticos de la organización, con el argumento de comenzar donde la creación de valor es mayor. Si la organización no tiene una curva de experiencia en la aplicación de BPMN, puede caer en la misma trampa que se conoce en todos los otros tipos de proyectos de TI, sean estos desarrollos de software, implementación de ERP o cualquier tipo de solución de negocio: Un proyecto largo, de muchas discusiones, se sobreponen los presupuestos de costo, tiempo y calidad. No le podemos recomendar este camino para introducir BPMN en su organización.

Propóngase mejor de empezar con un proceso simple, compacto y con el nivel descriptivo. Aprenda a definir el alcance (desde - hasta), alinear los objetivos, identificar los participantes, los subprocessos, diferenciar la lógica normal de la excepcional. A modo de ejemplo la compra de insumos de oficina es un proceso que

tiene de todo un poco, pero acotado. A un usuario se le acabó algún insumo, por ejemplo la tinta de la impresora. El debe solicitar un repuesto y tiene que dirigirse, primero al jefe y luego al departamento de compras. El departamento lo tiene en bodega o tiene que iniciar un proceso de compras. Luego se le entrega el producto al usuario (cliente) y quedará contento. En el nivel operativo experimente todos los detalles del proceso de la solicitud, aprobación, cotización, compra y entrega. Luego como analista tiene que tomar la decisión que parte del proceso será automatizada y modele un pool del sistema de workflow. Este modelo lo transfiere al nivel técnico y haga sus experiencias en la coordinación con los ingenieros de procesos, que van a implementar el workflow. De esta forma usted habrá pasado en poco tiempo por todo el ciclo y habrá automatizado un proceso. También tendrá una gran experiencia en la aplicación de BPMN y sabrá cómo abordar un proyecto un poco más complejo, sin caer en las trampas del BPMN y en general lo que significa introducir un pequeño BPM.

La lección que va a aprender es que también pequeños procesos tienen su complejidad y que el «diablo se esconde en el detalle». Luego de haber tenido esta experiencia se dará cuenta de la complejidad que pueda tener un proceso de la cadena de valor para una organización. ¿Cuantas veces mayor? Por experiencia propia sólo le podemos recomendar, empiece dando pequeños pasos.

7.2. Seleccione una Herramienta

«A fool with a tool is still a fool (inglés: «un tonto con una herramienta sigue siendo un tonto»). Este refrán es muy cierto y si bien se ha aplicado a muchas ciencias para el modelamiento de procesos, lo es más aun porque modelar hoy en día sin apoyo de una herramienta es absolutamente impensable.

Si usted quiere desarrollar su propio estilo en BPMN debería escoger una herramienta con el máximo de funcionalidades y posibilidades de configuración posible. Tiene que poder representar todos los símbolos de la paleta del nuevo BPMN, entonces cuales símbolos va a utilizar y cuales no, será su decisión, no la del fabricante. Le recomendamos que no haga una gran inversión al principio, pruebe una herramienta que le parezca y desarrolle su propio estilo. Es muy posible que más adelante quiera cambiarse de entorno. En este libro desarrollamos todos los diagramas con Microsoft Visio. Gracias a una comunidad de BPM que se formó en Berlín («Berliner BPM-Offensive»), se desarrolló una plantilla para la simbología de BPMN 2.0, de las cuales hoy en día existen muchas para Visio.

Con MS Visio puede diagramar lo que se le ocurra y los diagramas tienen una buena calidad de impresión, razón por la cual se utilizó mayormente para este trabajo, pero el lector tiene que tener claro que MS Visio no es una herramienta de modelamiento, sino una de diagramación. Le puede servir para ensayarse con algunos diagramas,

pero no para un proyecto real y menos si existen herramientas de modelamiento de software libre. Hoy en día existen cientos de herramientas para modelar en BPMN. La más conocida en región hispana es el modelador de Bizagi, que el usuario puede bajar libremente del sitio de la empresa.

Una buena alternativa para proyectos más complejos son las nuevas herramientas full web que cuentan con un repositorio integrado, que tienen versiones comunitarias (gratis) y que si les acomodan más adelante las pueden licenciar para su empresa. Al respecto le queremos presentar un proyecto de herramienta de modelamiento BPMN 2.0:

El sitio de Idungu (www.idungu.cl) presenta una herramienta full web de arquitectura empresarial que incluye un modelador de procesos para BPMN 2.0 y cuyo desarrollo es patrocinado por el BPM Center (www.bpmcenter.cl), del Departamento de Informática, de la Universidad Técnica Federico Santa María en Chile. La plataforma cuenta con una versión comunitaria en donde podrá registrarse y modelar procesos con un repositorio integrado.

Antes de iniciar un proyecto real le recomendamos que se entrene en BPMN y pruebe algunas herramientas. Cuando se sienta firme en la notación y haya desarrollado su propio estilo, debería formular sus requerimientos a una «verdadera» herramienta de BPMN. En el proceso de selección sea crítico, porque aún existen muchos proveedores que predicen ser compatibles con la notación, pero ¡no lo son! Revise si incluye toda la paleta de símbolos y funcionalidad de revisión de sintaxis, pero igualmente no debe imponerle de modelar de una cierta forma. ¡El analista decide! Otra funcionalidad muy importante es que cuente con un repositorio integrado o centralizado de los modelos. A continuación enumeraremos otras características importantes en proyectos reales:

Administración de Versiones (Versionamiento - Recuperación - Comparación)

Administración de Atributos

Documentación de Modelos

Chequeo de reglas sintácticas

Modo de publicación para usuarios (Poder agregar comentarios a los objetos y modelos)

Administración de Usuarios, Grupos y Perfiles

Conversión a imágenes

Generación de documentos Pdf, Word, PowerPoint

Animaciones, por ejemplo de Token

Metamodelo configurable

Exportación e importación de modelos a archivos tipo XML-BPMN, BPEL, XPDL

En general podemos segmentar el mercado en herramientas del tipo Business Process Analysis (BPA) y Modeler de BPM-Suite que también incluyen la Process Engine. Los proveedores de plataformas BPA no han sido consecuentes en apoyar la notación como la especificación lo exige y en muchas ocasiones incluso se salían del estándar, argumentando que sus ampliaciones eran mejores, pero la situación está cambiando, porque la nueva versión es mucho más amplia, poderosa y estricta al respecto. Los proveedores de BPM-Suite eran más consecuentes pero más restringidos, debido a la necesidad de generar modelos técnicos. A la mayoría de los proveedores de BPM-Suite les faltan funcionalidades importantes, por ejemplo son muy débiles en el apoyo de documentación para modelos de negocio y sobre todo en la falta de repositorios integrados. Consideramos un chiste de mal humor cuando los proveedores de BPMS argumentan que los proveedores de BPA, sólo cuentan con modeladores para diagramar, porque no incluyen una Process Engine. Al respecto: Modeladores sin un repositorio integrado son las verdaderas herramientas de diagramación, si contienen o no una Process Engine, no tiene relación para este tipo de clasificación.

Los modeladores pegados a los BPMS por lo general llevan al modelo técnico, debido a que la finalidad es la implementación. Existen algunos proveedores que ofrecen ambos entornos BPA y BPMS-Modeler. En este caso ambas plataformas se encuentran separadas y son integradas a través interfaces de integración.

Sin embargo el mundo de la tecnología se mueve más rápido de lo que nosotros pensamos. Siempre nacen nuevas iniciativas que nos sorprenden. Así por ejemplo la nueva plataforma de BPM open source que presenta la empresa Camunda (<http://camunda.org/>), una plataforma completa para BPM sobre BPMN 2.0 en modalidad opensource (Open Source BPM and Workflow with BPMN 2.0).

7.3. Agradecimientos al lector

Queremos agradecer al lector por haber depositado su confianza en nosotros y haberse tomado el tiempo de leer y estudiar esta obra. Esperamos que le sirva para mejorar los procesos de su organización o la de sus clientes. Si usted contribuye en mejorar la gestión de los procesos de su organización aplicando BPMN, estará automáticamente aportando a una mejor creación de valor de la organización hacia sus clientes. Entonces también habremos logrado el objetivo de nuestro libro.

Con el objetivo de seguir mejorando este trabajo en las próximas ediciones, le agradeceremos enviar su feedback a bernhard.hitpass@usm.cl.

Capítulo 8

BPMN Terminología Inglés-Español

Inglés Español

Abstract Process

Activity

Adhoc Subprocess

Annotation

Association (not directed/ directed/bidirectional) Attached Event

Proceso abstracto

Actividad

Subproceso ad hoc

Comentario

Asociación (no direccional/ direccional/bidireccional) Evento attachado

Cancel Event

Catching Event

Collaboration Process Collaborational Process Collapsed Subprocess Compensation

Event Complex Gateway

Conditional Event

Conditional Flow

Compensation

Evento de cancelación Evento de captura

Proceso de colaboración Proceso colaborador Subproceso cerrado

Evento de compensación Gateway complejo

Evento condicional

Flujo condicional

Compensación

259 Data Object

Data-based Exclusive Gateway Default Flow

End Event Error Event Event

Event-based Exclusive Gateway Gateway exclusivo basado en eventos Objeto de datos

Gateway exclusivo de datos (XOR) Flujo por defecto

Evento de término Evento por error Evento (ocurrencia)

Exception

Exception Flow
Expanded Subprocess Excepción
Flujo por excepción
Subproceso expandido (abierto)

Gateway Group
Gateway (compuerta) Agrupación

Inclusive Gateway Intermediate Event Gateway inclusivo (y/o) Evento intermedio

Lane
Link Event Loop
Loop Activity Lane (carril)
Evento de link (de conexión) Loop (bucles)
Actividad de loop

Message
Message Event Message Flow
Multiple Event Multiple Instance Mensaje
Evento de mensaje Flujo de mensaje Evento múltiple Instancia múltiple

Parallel Gateway Participant
Pool
Private Process Process
Property
Public Process Gateway paralelo (and) Participante
Pool (representa un proceso) Proceso privado (cerrado) Proceso
Propiedad
Proceso público (abierto)

Role Rule Rol
Regla 261

Sequence Flow Signal Event Start Event Subprocess
Flujo de secuencia Evento de señal Evento de inicio Subproceso

Task
Terminate Event Throwing Event Timer Event
Transaction
Trigger
Actividad (tarea)
Evento terminal (finaliza todas las marcas) Evento gatillador
Evento de tiempo (temporizador)
Transacción

Disparador

Bibliografía

[FreRue10] Freund, Jakob, y Rücker, Bernd (2010). Praxishandbuch BPMN 2.0. Hanser Verlag.

[FreundRuecker12] Freund, Jakob, & Rücker, Bernd (2012). Real-Life BPMN

[All08]

[Hitpass09]

[Hitpass12]

[HitpassLeiva12]

[SmiFin02]

[ABPMP09]

[DM08]

[Fis09] Allweyer, Thomas (2008). BPMN - Business Process Modeling Notation, Books on Demand.

Hitpass, Bernhard tión por Procesos cess Management (BPM), artículo BPM Center, (<http://www.bpmcenter.cl/index.php/publicaciones>)

Hitpass, Bernhard (2012). Business Process Management (BPM): Fundamentos y Conceptos de Implementación, BHH Ltda., Santiago de Chile, ISBN: 978-956-345-977-7

Hitpass, Bernhard & Leiva, Juan (2012). Modeling a Retailer B2B Integration Portal as Choreography Intermediary with BPMN 2.0 Choreography Diagrams, SCCC 2012: 247-252

Smith, Howard, Fingar, Peter (2002). Business Process Management (BPM): The Third Wave, Meghan-Kiffer Press ABPMP (2009). Guide to Business Process Management, Common Body of Knowledge, BPM CBOK, 2da. Edición, 2009
Decker, Gero, y Jan Mendling (2008). Process Instantiation. data and Knowledge Engineering (DKE). Volumne 68 Fischermanns, Guido (2009). Praxishandbuch Prozessmanagement. Schmidt (Götz), Wettenberg, Band 9
(2009). Introducción a Gesde Negocio, Business Pro264 BIBLIOGRAFÍA
[IDEF10] IDEF (2010). En <http://en.wikipedia.org/wiki/IDEF>, consultado diciembre 2010. [Obj09] Object Management Group cess Modeling Notation (2009). (BPMN) Business ProVersion 1.2, <http://www.omg.org/spec/BPMN/1.2/PDF>

[Obj11] Object Management Group (2011). cess Model and Notation (BPMN) <http://www.omg.org/spec/BPMN/2.0/PDF> Business ProVersion 2.0,

[SG06] Markus Schacher and Patrick Grössle (2006). Agile Unternehmen durch Business Rules. Springer Verlag

[BRG10] Comunidad para Reglas de Negocio: <http://businessrulesgroup.org/home-brg.shtml>, revisado: 24-01-14

[BRC10] Comunidad para Reglas de Negocio:<http://www.brcommunity.com/>, revisado: 24-01-14 [Senn92] Senn, James A. (1992). Análisis y Diseño de Sistemas de Información, McGrawHill, México, Segunda Edición Español

[Silver09] Silver, Bruce (2009). BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0, Cody-Cassidy Press

[Gad10] Gadatsch, Andreas (2010). Grundkurs GeschäftsprozessManagement, Vieweg+Teubner, Wiesbaden, Sexta Edición, Alemán

[HamCham93] Hammer,M, Champy,J. (1993). Reengineering the Corporation, New York

Índice de figuras

1. Popularidad de notaciones para modelar procesos	4
1.1. El ciclo de BPM	5
1.2. Automatización de un proceso con un Process Engine	9
1.3. Proceso simple en BPEL en forma gráfica y como XML	11
1.4. Matriz organizacional orientada a procesos	14
1.5. Marco estructural para BPMN (camunda BPMN-Framework)	16
2.1. Elementos básicos de BPMN	21
2.2. Nuestro primer proceso	25
2.3. XOR-Gateway o gateway exclusivo de datos	28
2.4. Símbolos equivalentes a XOR-Gateway	29
2.5. Modos de utilización de XOR-Gateway	30
2.6. Utilización de XOR-Gateway como a) XOR-Split y XOR-Join de forma compacta y b) XOR-Split y XOR-Join de forma separada	30
2.7. Preparación de ensalada y plato de fondo	31
2.8. Preparación de ensalada junto con plato de fondo	32
2.9. Caso And-Split sin And-Join	33
2.10. Revisión de tiempo de duración de instancias con AND-Split	34
2.11. Diferentes opciones en la selección de nuestra cena	35
2.12. Utilización de OR-Gateway como OR-Split y OR-Join	35
2.13. Revisión de tiempo de duración de instancias con OR-Gateway en el OR-Join	36
2.14. Una versión compacta con muchas opciones	37
2.15. Flujo por defecto	38
2.16. Pedido de pizza con un AND-Join	39
2.17. Pedido de pizza con un XOR-Join	39
2.18. Pedido de pizza con un OR-Join	39
2.19. Pedido de pizza con un Complex-Join	40
2.20. Con Gateway complejos se pueden representar también casos como «M out of N-Joins»	40
2.21. Equivalencia entre a) utilización de flujos condicionales y b) utilización de OR-Split y XOR-Join	42
2.22. XOR-Gateways combinados	43
2.23. Similitud entre a) utilización de AND-Gateway y XOR-Gateway b) utilización de flujos condicionales	44
2.24. Utilización de Lanes	45
2.25. Estructura jerárquica de lanes	46
2.26. Uso incorrecto de actividad en Lanes	46
2.27. Uso correcto de actividad en lanes	47
2.28. Estructura de lane prohibida a partir de la versión BPMN 2.0	47
2.29. Un Evento gatillante de proceso	49
2.30. Dos eventos que	

independientemente pueden gatillar un proceso (evento 1 o evento 2)	49
2.31. Dos eventos que producirían un punto muerto (deadlock) en interpretadores de BPMN	49
Utilización de evento intermedio	50
dos eventos intermedios continuos	51
2.34. Utilización de AND-Gateway con eventos intermedios	51
2.35. Utilización de evento intermedio y sobrepuerto (attached)	52
2.36. Utilización de evento intermedio y sobrepuerto del tipo "no interrupción"	52
2.37. Tipos de eventos de mensajería	53
2.38. Pedido y recepción de pizza como eventos de mensaje	54
2.39. Caso de incorrecta utilización de actividad con evento intermedio .	54
2.40. Utilización de evento intermedio y sobrepuerto del tipo "mensaje" .	55
2.41. Caso de correcta utilización de Actividad (para envío) y Evento (para recepción)	55
2.42. Tipos de Eventos de Tiempo	56
2.43. Utilización de eventos de tiempo	57
2.44. El timeout para la actividad seleccionar pizza es de 30 minutos . .	57
2.45. Utilización de evento de tiempo, intermedio y sobrepuerto del tipo "no interrupción"	58
2.46. Tipos de Eventos de Error	58
2.47. Tipos de Eventos Condicionales	59
Proceso de hornear pizza con eventos condicionales	59
2.49. Proceso de hornear pizza con evento condicional y temporal . .	59
2.50. Tipos de Eventos de Señal	60
2.51. Caso de correcta utilización de Evento de Señal	60
2.52. Caso de correcta utilización de evento de término	61
2.53. Caso de correcta utilización de Evento Terminador	61
2.54. El evento terminador	62
2.55. Equivalencia entre a) utilización de Eventos de Conexión y b) utilización de flujo de secuencia	62
2.56. Utilización de Eventos de Conexión	63
2.57. Tipos de eventos de compensación	63
2.58. Organización de una salida para el día viernes	65
2.59. Organización de una salida para el día viernes con eventos de compensación . .	66
2.60. Tipos de Eventos Múltiple	67
2.61. Caso de correcta utilización de Evento Múltiple	67
2.62. Un modelo alternativo al proceso de la figura 2.61	68
2.63. Tipos de eventos de múltiple paralelo	68
2.64. Tipos de eventos de escalación	69
2.65. Caso con utilización de XOR-Gateway basado en datos	70
2.66. Caso con uso deficiente de evento intermedio	70
2.67. Caso de pedido de pizza con Event-Gateway	71
2.68. Posibles eventos ante los cuales puede reaccionar el Event-Gateway .	71
2.69. Equivalencia entre a) Dos eventos que independientemente pueden gatillar un	

proceso y b) Evento múltiple como evento de inicio para gatillar un proceso	72
2.70. Desigualdad entre a) Dos eventos que independientemente pueden gatillar un proceso y b) Evento múltiple paralelo como evento de inicio para gatillar un proceso	73
2.71. Simbología para tipos de actividades en BPMN 2.0	76
2.72. Caso con uso de actividad con propiedad de loop	76
2.73. Cuatro casos que representan el mismo proceso de la figura 2.72. a) Alternativa 1 con utilización de XOR-Join y XOR-Split b) Alternativa 2 con utilización de XOR-Split c) Alternativa 3 con utilización de XOR-Join y flujos condicionales d) Alternativa 4 con utilización de flujos condicionales	77
2.74. Caso con uso de actividad múltiple	78
2.75. Casos con uso de Actividad de compensación a) Compensación de loop manual b) Compensación múltiple manual	79
2.76. Actividades globales segun BPMN 2.0	80
2.77. Una actividad y un subprocesso	81
2.78. Caso con representación de subprocesso en un diagrama propio por separado . .	82
2.79. Caso con representación de subprocesso en el mismo diagrama principal del proceso	82
2.80. Caso con posicionamiento de eventos de inicio y término en los límites del subprocesso	83
2.81. Equivalencia entre a) Caso con utilización de AND-Gateway y b) Caso con uso de subprocesso	84
2.82. Ejemplos para la reutilización de subprocessos	86
2.83. Caso con evento sobrepuerto que interrumpe subprocesso completo	87
2.84. Caso con subprocesso que comunica un error al proceso superior	88
2.85. Caso con Evento de Escalación que comunica un Mensaje al Proceso Superior	89
2.86. Equivalencia entre a) Subprocesso con propiedad de loop y b) Flujo con actividades y XOR-Gateway que se comportan como loop	90
2.87. Caso con uso de subprocesso con propiedad Ad-Hoc	90
2.88. Caso con uso de subprocesso con propiedad Ad-Hoc con poca estructuración interna	91
2.89. Caso con uso de subprocesso expandido del tipo Transacción	92
2.90. Ejemplos de subprocessos impulsados por eventos	93
2.91. Caso con uso de subprocessos impulsados por eventos	94
2.92. Caso con utilización de Eventos sobrepuertos en el subprocesso como modelamiento alternativo a figura 2.90	95
2.93. Actividades y responsables de actividades	96
2.94. Flujo sin nombre de Pool	97
2.95. Flujo con cuatro participantes en Pools propios	97
2.96. Elementos permitidos para el intercambio de información (flujos de mensajes) entre pools	99
2.97. Elementos no permitidos relacionados con flujos de mensajes	99
2.98. El proceso de pedido de pizza desde el punto de vista del cliente	100
2.99. La entrega de pizza	

desde el punto de vista del proveedor	100
2.100.El proceso de negocio de pedido y entrega de pizza representado en un pool	101
2.101.El proceso de negocio del pedido y entrega de pizza modelado en dos pools	101
2.102.El proceso de negocio de pedido y entrega de pizza modelado correctamente	102
2.103.Caso con uso de Pool abierto y Pool cerrado	103
2.104.Intercambio de información entre dos pools cerrados	104
2.105.Caso con uso de Pool cerrado con propiedad de instancias múltiples	104
2.106.Ejemplo de uso con objetos de datos	106
2.107.Nuevos símbolos de datos en BPMN 2.0	107
2.108.Caso con utilización de Comentarios y Agrupaciones	109
2.109.Artefactos propios	110
2.110.Proceso de Compra de Pizza como a) Modelo de Conversación b) Modelo de Coreografía y c) Modelo de Colaboración	112
2.111.Modelo de Coreografía en BPMN2.0 para el Proceso de Compra y Recepción de Mercadería vía B2B (Inbound Logistics)	114
2.112.Objetos del Diagrama Análisis Estructurado según Yourdon o Gane y Sarson	117
2.113.Descomposición jerárquica de un diagrama de análisis estructurado .	118
2.114.Ejemplo Diagrama del Contexto	118
2.115.Clasificación de algunas técnicas de diagramación para modelamiento de procesos	121
2.116.Mapeo de EPC hacia BPMN	127
2.117.Mapeo de UML diagrama de actividades a BPMN	128
2.118.Ejemplo de representación de indicadores en BPMN	130
 3.1. Nivel 1 Descriptivo del Marco Estructural para BPMN	132
3.2. Modelos del primer nivel se levantan a partir de dos situaciones . . .	135
3.3. Modelo de proceso de contratación de personal para nivel descriptivo	139
3.4. Modelo de proceso de contratación de personal con Lane de área convertido en Pool	140
3.5. Modelo de proceso de contratación de personal con cada participante en Pool propio	141
3.6. Caso de modelo de colaboración utilizando Pool cerrado	142
3.7. Modelo de proceso de contratación de personal con actividades y subprocesos en nivel descriptivo	144
3.8. Recomendaciones de uso de Gateways en el nivel descriptivo	146
3.9. Modelo de proceso de contratación de personal sin eventos de inicio y de término en nivel descriptivo	147
3.10. Definición de hitos en el proceso de contratación de personal	149
3.11. Modelo de proceso de contratación de personal con objetos de datos en nivel descriptivo	150
3.12. Modelo de proceso de contratación de personal con artefactos propios en nivel	

descriptivo	151
3.13. El proceso de contratación de personal como ejemplo de nivel descriptivo	152
3.14. La fase de análisis en el ciclo de BPM	154
3.15. Cadena causal como resultado del taller de análisis	156
3.16. Diagrama de conversación en dos niveles de descomposición para proceso de contratación de personal	158
3.17. Diagrama de coreografía para proceso de contratación de personal .	159
4.1. Nivel 2 Operativo del Marco Estructural para BPMN	162
4.2. Roles y vistas de los participantes en un proyecto de BPM	164
4.3. Modelo de proceso de contratación de personal para nivel descriptivo	167
4.4. Modelo que muestra el comienzo del traspaso al nivel 2	169
4.5. Proceso de licitación de cargo desde la perspectiva del participante jefe de área	170
4.6. Proceso de cicitación de cargo según normas de BPMN 2.0	171
4.7. Proceso de licitación de cargo desde la perspectiva del participante usuario de RRHH	171
4.8. Diagrama de colaboración para proceso de licitación de cargo	173
4.9. Diagrama de coreografía para proceso de licitación de cargo	173
4.10. Transición del nivel 2 operativo al nivel 3a técnico	175
4.11. Proceso de licitación de cargo apoyado por Process Engine para área de negocio	176
4.12. Proceso de licitación de cargo apoyado por Process Engine para área de RRHH	177
4.13. Representación del proceso de licitación en un sistema de workflow .	178
4.14. Requerimientos adicionales para la implementación técnica	181
4.15. Prototipo de pantallas para el proceso de licitación de cargo	182
4.16. Transición del nivel 2 operativo al nivel 3b técnico	184
4.17. Diagrama de flujo simple par FPY en BPMN	186
4.18. El proceso Orden de Compra en el «Happy Path»	188
4.19. Alternativas para representar problemas con BPMN	189
4.20. Alternativas de representación de errores desviadas del «Happy Path»	190
4.21. Caso con uso de subprocesso con propiedad Ad-Hoc	193
4.22. Proceso Orden de Compra con revisión de credibilidad	194
4.23. Proceso Orden de Compra con regla de negocio modelada	195
4.24. Tabla de decisión para reglas de negocio	196
4.25. Forma de representar reglas de negocio en modelos de BPMN	197
4.26. Actividad tipo regla de negocio	198
4.27. Evento de condición y Motor de Reglas	199
4.28. Frecuencia de uso de objetos BPMN en el nivel operacional	201
5.1. Nivel 3 Modelo Técnico del marco estructural para BPMN	204

5.2. Forma de trabajo de un Process Engine	207
5.3. El circulo mágico de la BPM-Suite	209
5.4. Modelo de proceso de licitación de cargo para nivel técnico	213
5.5. Llamada de servicios síncrono y asíncrono en BPMN	216
5.6. El modo asíncrono o síncrono no es siempre igual en el nivel de negocio y el técnico	217
5.7. Modelo para creación de una instancia de proceso con evento de tiempo	219
5.8. Modelo para creación de una instancia de Proceso impulsada por componente de TI externa	220
5.9. Ejemplo de un Paralel-Event-Gateway como inicio múltiple	221
5.10. Diferentes patrones para caso de eventos de inicio múltiple	221
5.11. Eventos intermedios en el modelo técnico: caso a) con inconsistencia y caso b) con trato excepcional	222
5.12. Utilización de evento intermedio de condición en a) modelo de negocio y b) modelo técnico	223
5.13. Evaluación de condiciones a) con implementación de tecnología externa y b) sin implementación de tecnología externa	224
5.14. Ruteo técnico de Tokens aplicado a Gateways Paralelo y Exclusivo	226
5.15. Gateway Inclusivo (OR)	227
5.16. El comportamiento de espera del OR-Gateway	227
5.17. Gateway Complejo	228
5.18. Caso de proceso que se inicia con Event-Paralel-Gateway y finaliza con evento de término normal	229
5.19. Eventos de término clasificados en normal y anormales	230
5.20. Transacciones de negocio y técnicas	232
5.21. Ejemplo de compensación de un llamado de servicio	232
5.22. Subprocesos, Scope y vista de datos	234
5.23. Tratamiento de errores en instancias múltiples	235
 6.1. Reingeniería - Rediseño - Mejora Continua	241
6.2. Análisis de estructura según Bleicher (parte 1)	242
6.3. Análisis de estructura según Bleicher (parte 2)	243
6.4. Análisis de tiempo de ciclo	244
6.5. Análisis de responsabilidades: Antes de la optimización	249
6.6. Análisis de responsabilidades: Despues de la optimización	251

Índice alfabético

- Actividad de compensación, 78
- Actividad múltiple, 78
- Actividades Especiales, 73
- Actividades globales, 79
- Análisis de ciclo, 244
- Análisis de costeo de actividades, 247 Análisis de estructura, 241
- Análisis de responsabilidades, 248 Artefactos, 108
- Automatización de los Procesos, 8, 174

- BPA Suite Idungu, 255
- BPEL, 10, 205, 210, 212
- BPM, 1
- BPM Center, 255
- BPM Plataforma Camunda.org, 257 BPM-Governance, 240
- BPMN 1.2 (Business Process Modeling Notation), 11
- BPMN 2.0 (Business Process Model and Notation), 12
- BPMN-Framework, 16
- Business Activity Monitoring (BAM), 10 Business Process Reengineering (BPR), 238

- Ciclo BPM, 5
- Colaboración, 99, 105, 111, 115, 140, 149, 165, 172, 177, 186
- Compensación, 63
- Comunicación asíncrona, 216
- Comunicación síncrona, 216 Conversación, 111
- Coreografía, 111, 158, 173
- Correlación, 24, 225
- Correlación de contexto, 225 Correlación de llaves, 225
- Critical Path Method (CPM), 247 Cuello de botella, 245

- Ejecución de modelos, 208
- Elementos básicos del BPMN, 20 Errores, 187
- Escurrimiento, 223
- Event Driven Architecture (EDA), 224 Event driven Process Chain (EPC, 124
- Eventos, 26, 48
- Expression Language, 215

Expressions, 214

Filtros, 144, 199

First Pass Yield, 185

Flujo condicional, 41

Flujo de Procesos sin Gateways, 41 Flujo de Secuencia, 27

Flujo por defecto, 37

Flujos de Mensajes, 95

Forward Engineering, 211

Gateway complejo, 38

Gateway exclusivo, 27

Gateway inclusivo, 34

272 ÍNDICE ALFABÉTICO 273

Gateway paralelo, 30 Gateways, 27

Happy path, 185

Herramientas de modelado para BPMN,
254

Process Engineer, 15

Process Execution, 207, 213 Process Manager, 14

Process Owner, 13

Process Participant, 14

Propiedades de actividades, 76

IDEF, 120

Indicadores, 129

Instance Routing, 225

Instancia de proceso, 23, 228 Instancias múltiples, 234 Integración de datos, 183

Interfaz de usuario, 183 Invocación de servicios, 216

Lanes, 44

Log files, 208 Loop, 76, 234

Mejora, 240

Model Driven Architecture (MDA), 211 Modelamiento explícito de Errores, 187

Modelo de Procesos Descriptivos, 131 Modelo de Procesos Operacional, 161

Modelo de Procesos Técnico, 203, 213 Rediseño de procesos, 239 Reglas de negocio, 182, 194 Reglas de ruteo, 195

Reingeniería de procesos, 238 Roundtrip Engineering, 211

Script, 75

Semántica de Ejecución, 218 Structured Analysis, 116 Subprocesos, 80

Técnicas de Análisis y Mejora, 237 Tabla de decisión, 196
Tiempo de ciclo, 31, 60, 244 Tiempo de espera, 244
Tiempo de valor agregado, 244 Tipos de actividades, 73
Token, 23
Transacción, 92, 231

Nivel 1 - Procesos Descriptivos, 17 Nivel 2 - Procesos Operacionales, 17 Nivel 3 -
Modelo Técnico, 17
UML, 125

Vistas de un proceso en BPMN, 22

Object Management Group (OMG), 11 Objetos de Datos, 105
Orquestación, 96, 98, 165, 174, 183, 234

Participantes en un proceso, 13 Pool con instancias múltiples, 104 Pools cerrados,
102

Procesos de los Participantes, 169 Process Analyst, 14

Process Engine, 8, 177, 206 XPath Expression, 215 XPDL, 205