

DESCUBRIMIENTO AUTOMÁTICO DE PROCESOS DE NEGOCIO BASADO EN SEMÁNTICA DE COMPORTAMIENTO



CRISTHIAN NICOLÁS FIGUEROA MARTÍNEZ

ANEXOS

Director:

Juan Carlos Corrales

Doctor en Ciencias de la Computación

Universidad del Cauca

Facultad de Ingeniería en Electrónica y Telecomunicaciones

Departamento de Telemática

Línea de Investigación en Aplicaciones y Servicios sobre Internet

Popayán, noviembre de 2011

ANEXOS

Contenido

Lista de Figuras.....	145
Lista de Tablas.....	148
A Patrones de Control de Flujo.....	151
A.1 Introducción.....	151
A.2 Clasificación de los Patrones de Control de Flujo	151
A.3 Implementación de los patrones en BPMO.....	153
A.3.1 Viabilidad de Implementación de los Patrones en BPMO.....	154
A.4 Descripción de los Patrones Seleccionados	156
A.4.1 Patrón Secuencia (Sequence).....	156
A.4.2 Patrón División Paralela (Parallel Split)	156
A.4.3 Patrón Sincronización (Synchronize).....	157
A.4.4 Patrón Selección Exclusiva (Exclusive Choice)	157
A.4.5 Patrón Combinación Simple (Simple Merge).....	158
A.4.6 Patrón Selección Múltiple (Multiple choice)	158
A.4.7 Patrón Combinación Múltiple (Multi Merge).....	159
A.4.8 Patrón Selección Diferida (Deferred Choice).....	160
A.4.9 Patrón Discriminante (Discriminator)	160
A.4.10 Patrón de enrutamiento paralelo intercalado (Interleaved Parallel Routing):.....	161
A.4.11 Patrón de Combinación Múltiple Sincronizada (Multi Merge Synchronize) 161	
A.4.12 Patrón de Instanciación Múltiple (Multiple Instantiation).....	162
A.5 Comportamiento y Funcionalidad de los Patrones.....	162

A.5.1	Clasificación de patrones según su funcionalidad	164
A.5.2	Clasificación de patrones según su comportamiento	164
A.6	Relaciones entre los patrones de control de flujo	166
A.6.1	Relación de especialización entre patrones.....	166
A.6.2	Relación de composición entre patrones	168
A.7	Ontología de Patrones de Control de Flujo	172
B	Lenguajes Semánticos y Ontologías de Dominio	179
B.1	Introducción.....	179
B.2	Lenguajes de Descripción de Ontologías	179
B.2.1	RDF (Resource Description Framework)	180
B.2.2	RDFS (RDF Schema)	181
B.2.3	OWL (Web Ontology Language).....	182
B.2.4	WSML (Web Service Modeling Language)	182
B.2.5	Selección del lenguaje semántico de descripción de ontologías.	184
B.3	Entorno de Modelado de Procesos de Negocio BPMO.....	185
B.4	Ontologías del Dominio de las Telecomunicaciones	194
B.4.1	eTOM (enhanced Telecom Operations Map).....	196
B.4.2	SID (Shared Information/Data Model).....	199
B.5	Adaptación de Ontologías del Dominio de las Telecomunicaciones	201
C	Enriquecimiento Semántico de Procesos BPMO.....	203
C.1	Introducción.....	203
C.2	Definición del Caso de Estudio y Modelamiento del Proceso.....	204
C.3	Enriquecimiento Semántico de entradas/salidas de las Tareas del Proceso de Negocio	206
C.4	Enriquecimiento Semántico de los nombres de las Tareas.....	210
D	Teoría de Grafos	217
D.1	Introducción.....	217
D.2	Definiciones Básicas	217
D.3	Clasificación de Grafos.....	219
D.3.1	Grafo regular.....	221
D.3.2	Grafo bipartito	221
D.3.3	Grafo completo	221
D.3.4	Grafo nulo	222
D.3.5	Grafos Isomorfos	222

D.3.6	Grafos Conexos:.....	223
D.3.7	Árboles	223
D.4	Recorrido de un Grafo.....	224
D.4.1	Recorrido en anchura	224
D.4.2	Recorrido en profundidad	225
D.5	Isomorfismo de Grafos	226
E	Repositorios de Procesos de Negocio	229
E.1	Introducción.....	229
E.1.1	Repositorios de Servicios Web	229
E.1.2	Repositorios de Procesos de Negocio	230
F	Colección de Procesos de Negocio de Prueba	239
F.1	Procesos de Negocio de Consulta	239
F.2	Procesos de Negocio de Comparación	241
G	Plataforma para Evaluar Sistemas de Recuperación de Procesos de Negocio 253	
G.1	Introducción.....	253
G.2	Descripción de la Plataforma	254
G.3	Diagrama en Capas	255
G.3.1	Capa de aplicación	256
G.3.2	Capa de mediación.....	261
G.3.3	Capa de fundamentación:.....	261
H	BeMantics: Una Plataforma Recuperar Procesos de Negocio basada en Semántica del Comportamiento	263
H.1	Introducción.....	263
H.1.1	Diagrama de Paquetes del Analizador Estructural.	263
I	Resultados de los Análisis de Rendimiento y Calidad.....	269
I.1	Introducción.....	269
I.2	Análisis de Rendimiento.....	269
I.2.1	Análisis de Rendimiento del Repositorio.....	270
I.2.2	Análisis de Rendimiento del Analizador Estructural y Semántico	273
I.3	Análisis de Calidad.....	275
I.3.1	Análisis de Calidad del Repositorio.....	277
I.3.2	Análisis de Calidad del Analizador Estructural y Semántico	279
	Referencias	283

Lista de Figuras

Figura A.3-1: Ejemplo de implementación (Structured-Partial-Join).....	154
Figura A.3-2: Viabilidad de implementación de los patrones en la notación BPMO.	155
Figura A.4-1: Patrón Secuencia con tres tareas.....	156
Figura A.4-2: Patrón División Paralela con tres tareas.....	157
Figura A.4-3: Patrón Sincronización con tres tareas.	157
Figura A.4-4: Patrón Selección Exclusiva con tres tareas.....	158
Figura A.4-5. Patrón Combinación Simple con tres tareas.....	158
Figura A.4-6. Patrón Selección Múltiple con tres tareas.....	159
Figura A.4-7. Patrón Combinación Múltiple con tres tareas.	159
Figura A.4-8. Patrón Selección Diferida con tres tareas.....	160
Figura A.4-9. Patrón Discriminante con tres tareas.....	160
Figura A.4-10: Patrón de enrutamiento paralelo intercalado con 3 tareas.	161
Figura A.4-11. Patrón de Combinación Múltiple Sincronizada con 3 tareas.	161
Figura A.4-12. Patrón de Múltiple Instanciación con tres tareas.	162
Figura A.5-1: Patrones representados con (a) BPMO y (b) grafos.....	163
Figura A.6-1: Relación de especialización entre los patrones Parallel Split y Multiple Choice.	167
Figura A.6-2. Relación de composición entre los patrones Parallel Split, Exclusive Choice y Multiple Choice.....	169
Figura A.6-3. Relaciones de especialización y composición entre patrones de control de flujo.....	171
Figura A.6-4. Relaciones de especialización y composición para los patrones básicos de BPMO.....	172
Figura A.7-1. Ontología de patrones de control de flujo básicos.	174
Figura B.2-1: Variantes de WSML.....	184
Figura B.3-1: Modelador WSMO Studio.	186
Figura B.3-2: Representación gráfica del ejemplo “ <i>Develop and Train Workforce</i> ” descrito en WSML a partir del BPMO Modeler 1.4.....	187
Figura B.4-1: Mapa de Operaciones de Telecomunicaciones Mejorado – eTOM. ...	197
Figura B.4-2. Modelo de Información/Datos Compartidos – SID.....	200
Figura C.2-1: Subproceso de Autenticación – DAM en WSMO Studio.	205
Figura C.2-2: <i>Perspectiva</i> BPMO – WSMO Studio.....	206
Figura C.3-1: Adición de Entradas a las Tareas BPMO.	207
Figura C.3-2: Cuadro de Dialogo Nueva Entrada.....	207

Figura C.3-3: Visualización de entradas de la Tarea.....	208
Figura C.3-4: Propiedad Data Type - Entrada CustomerLogin.....	208
Figura C.3-5: Selección de Data Type - Entrada CustomerLogin.....	208
Figura C.3-6: Visualización de salidas de la Tarea.....	209
Figura C.3-7: Selección de Concept Type - Salida CustomerProfile.....	210
Figura C.4-1: Ontología del Proceso de Negocio de Autenticación en WSMT.....	211
Figura C.4-2: Definición de la variante de WSML.....	212
Figura C.4-3: Adición de ontologías de dominio.....	213
Figura C.4-4: Adición manual de anotaciones semánticas a la tarea Get Profile....	214
Figura C.4-5: Resultado de la consulta sobre la ontología del proceso.....	215
Figura D.2-1: Ejemplo de un grafo de proceso.....	218
Figura D.3-1: Ejemplos de grafos dirigidos y no dirigidos.....	220
Figura D.3-2: Grafos regulares y Grafo no regular.....	221
Figura D.3-3: Grafos bipartito y no bipartito.....	221
Figura D.3-4: Grafos completos.....	222
Figura D.3-5: Grafo nulo.....	222
Figura D.3-6: Grafos Isomorfos.....	223
Figura D.3-7: Grafo Conexo.....	223
Un árbol se define como un tipo de grafo que no contiene ciclos, es decir un grafo acíclico y conexo. En la Figura D.3-8, se observan ejemplos de árboles, como se puede notar ninguno de ellos contiene repeticiones (ciclos).	223
Figura D.3-9: Árboles.....	224
Figura D.4-1: Recorrido en profundidad y en anchura.....	225
Figura D.5-1: Ejemplo grafo isomorfo (a) y sub-grafo isomorfo (b).....	227
Figura G.3-1: Diagrama en Capas para la Plataforma para Evaluar Sistemas de Recuperación de Procesos de Negocio.....	256
Figura G.3-2: Página inicial para el Administrador.....	257
Figura G.3-3: Interfaz para gestionar Procesos.....	258
Figura G.3-4: Interfaz de visualización de procesos.....	259
Figura G.3-5: Interfaz principal para la comparación de procesos.....	260
Figura G.3-6: Interfaz principal de los resultados.....	260
Figura H.1-1. Diagrama de paquetes del analizador Estructural.....	264
Figura H.1-2: Diseñador de Procesos de Negocio WSMO Studio.....	265
Figura H.1-3: Vista de la versión Web del Repositorio.....	266
Figura H.1-4: Ejemplo de un resultado del proceso de correspondencia estructural y semántica.....	267
Figura I.2-1: Tiempo de almacenamiento vs Número de nodos (Nodes) de los procesos de negocio.	271
Figura I.2-2: Tiempo de Almacenamiento vs Número de Aristas de los procesos de negocio.....	272
Figura I.2-3: Tiempo de recuperación por cada consulta.....	273
Figura I.2-4: Número de procesos de negocio clasificados vs tiempo de recuperación.....	273
Figura I.2-5: Tiempo de correspondencia vs número de aristas.....	274
Figura I.2-6: Tiempo de correspondencia vs número de nodos.....	274

Figura I.3-1: Resultados Precisión y Exhaustividad Estructural para cada consulta del repositorio.....	277
Figura I.3-2: Resultados Precisión y Exhaustividad Lingüística para cada consulta del repositorio.....	278
Figura I.3-3: Resultados Precisión y Exhaustividad para Comportamiento Secuencial para cada consulta del repositorio.....	279
Figura I.3-4: Precisión y Exhaustividad Estructural para cada consulta del analizador estructural y semántico	280
Figura I.3-5: Precisión y Exhaustividad Lingüística para cada consulta del analizador estructural y semántico	281
Figura I.3-6: Precisión y Exhaustividad para el Comportamiento Secuencial para cada consulta del analizador estructural y semántico	282

Lista de Tablas

Tabla A.5-1. Clasificación en funcionalidad y comportamiento de los patrones.	165
Tabla A.6-1: Relaciones de especialización entre patrones de control de flujo.	168
Tabla A.6-2. Relaciones de composición entre patrones de control de flujo	170
Tabla A.7-1: Ontología de Patrones de Control de Flujo en formato WSML	178
Tabla B.3-1: Contenido del archivo del Modelo BP del ejemplo “ <i>Develop and Train Workforce.wsmf</i> ”	194
Tabla F.1-1: Descripción de los procesos de negocio de consulta y su dominio de aplicación.....	240
Tabla F.1-2: Características de los procesos de negocio de consulta.....	240
Tabla F.2-1: Descripción de los procesos de negocio del repositorio y su dominio de aplicación.....	249
Tabla F.2-2: Características de los procesos de negocio del repositorio.	252
Tabla I.2-1: Características del servidor de prueba	270
Tabla I.2-2: Tiempo de almacenamiento vs Número de nodos de los procesos de negocio.....	270
Tabla I.2-3: Tiempo de Almacenamiento vs Número de Aristas (Edges) de los procesos de negocio	271
Tabla I.2-4: Tiempo de recuperación y número de procesos de negocio clasificados por cada consulta	272
Tabla I.2-5: Tiempo de correspondencia vs número de aristas.....	273
Tabla I.2-6: Tiempo de correspondencia vs número de nodos.....	274
Tabla I.3-1: Procesos de negocio de consulta.....	275
Tabla I.3-2: Criterios analizados en las evaluaciones manuales	275
Tabla I.3-3: Criterios evaluación Manual vs Criterios evaluación Automática.....	276
Tabla I.3-4: Precisión y Exhaustividad Estructural para cada consulta del repositorio	277
Tabla I.3-5: Resultados Precisión y Exhaustividad Lingüística para cada consulta del repositorio.....	278
Tabla I.3-6: Resultados Precisión y Exhaustividad para Comportamiento Secuencial para cada consulta del repositorio.....	278
Tabla I.3-7: Precisión y Exhaustividad Estructural para cada consulta del analizador estructural y semántico.....	279

Tabla I.3-8: Precisión y Exhaustividad Lingüística para cada consulta del analizador estructural y semántico	280
Tabla I.3-9: Precisión y Exhaustividad para el Comportamiento Secuencial para cada consulta del analizador estructural y semántico	281

Anexo A

A Patrones de Control de Flujo

A.1 Introducción

En este anexo se presenta una revisión de los patrones de control de flujo y se define un conjunto de doce patrones que se pueden implementar en BPMO como lo requería la presente propuesta. Se realiza una clasificación de los patrones, se estudian sus relaciones y basado en ellas se propone una abstracción a través de una ontología de patrones de control de flujo descrita en el lenguaje WSML.

A.2 Clasificación de los Patrones de Control de Flujo

La clasificación de los patrones de control, que se presenta en esta sección, está basada en el trabajo de Russell et al [1], en el cual se propone la siguiente clasificación.

- *Patrones de Control de Flujo Básicos:* Esta clase de patrones captura los aspectos elementales del control de flujo de proceso y son similares a las definiciones de estos conceptos inicialmente propuestos por la WfMC [2].
- *Patrones de control de flujo de ramificación y sincronización avanzada:* Estos patrones se caracterizan por utilizar conceptos más complejos de

ramificación y sincronización que surgen a partir de los procesos de negocio.

- *Patrones de control de flujo de múltiples instancias:* Describen las situaciones en donde hay múltiples hilos de ejecución activos en un modelo de proceso que se relaciona a una misma actividad (por lo tanto, comparten la misma definición de la implementación). Las múltiples instancias pueden ser declaradas en 3 situaciones:
 - Una actividad puede iniciar múltiples instancias de sí misma cuando se ha activado (se denota esta clase de actividad como una actividad de múltiples instancias).
 - Una actividad dada se activa en múltiples ocasiones como una consecuencia de recibir activaciones independientes (por ejemplo una parte de un ciclo o una instancia de un proceso en el cual hay muchos hilos de ejecución concurrentes como un resultado de un patrón que une el control de flujo).
 - Dos o más actividades en un proceso comparten la misma definición de implementación. Esto podría ser la misma definición de actividad, como en el caso de múltiples instancias de actividades o una definición de un subproceso común en el caso de un bloque de actividades. Dos (o más) de estas actividades se inician de tal forma que sus ejecuciones se superponen ya sea parcial o totalmente

Aunque estas 3 situaciones potencialmente involucran múltiples instancias concurrentes de una actividad o un subproceso, la primera de ellas atrae mayor interés, ya que ellas requieren las activaciones y sincronización de múltiples instancias de actividades concurrentes. Este grupo de patrones se enfoca en las diferentes maneras en las cuales los eventos pueden ocurrir

- *Patrones Basados en Estados:* Estos muestran la situación donde las soluciones son más fácilmente ejecutadas en los lenguajes de procesos que soporten la noción del estado. En este contexto se consideran los estados de

una instancia de un proceso, para incluir una amplia colección de datos asociados con la ejecución actual incluyendo el estado de varias actividades, así como también la información relevante al proceso, tal como los elementos de las actividades y los elementos de datos.

- *Patrones de Cancelación y Terminación Forzada:* Existen patrones que tienen variantes que utilizan el concepto de cancelación de actividades, donde las instancias de actividades son activadas o desactivadas. Varias formas de excepción manejadas en los procesos son también basadas en conceptos de cancelación
- *Patrones de Iteración:* Capturan un comportamiento repetitivo en un proceso de negocio.
- *Patrones de Terminación:* Estos abordan las circunstancias sobre las cuales un proceso de negocio se considera como completo
- *Patrones de Activación:* Estos se encargan de manejar las señales externas que pueden ser requeridas para iniciar ciertas actividades.

A.3 Implementación de los patrones en BPMO.

En el trabajo de Rusell et al [1] se presenta además de los patrones de control de flujo, un estudio acerca de la compatibilidad de los patrones con las herramientas y lenguajes de modelado de procesos de negocio. Sin embargo, no se considera el lenguaje BPMO con el cual se trabaja en el presente trabajo de grado; por lo tanto en esta sección se hace una evaluación de la viabilidad de la implementación de los patrones de control de flujo en la notación BPMO. Para esto se tuvo en cuenta la funcionalidad de cada patrón.

Por ejemplo, un patrón descrito en [1] denominado “*Structured-Partial-Join*” y cuya funcionalidad consiste en generar un flujo de salida únicamente cuando se cumpla un subgrupo determinado de actividades dentro de un grupo total, es decir, en este caso el elemento “2-out of-3-join” solamente genera un flujo de salida cuando se cumple un subgrupo de 2 actividades de un total de 3 (Figura A.3-1). El elemento

“*Structured-Partial-Join*” no existe directamente dentro de los elementos de BPMO, pero analizando su funcionamiento se puede encontrar un equivalente por medio de una relación de composición a partir de los elementos de BPMO (Figura A.3-1).

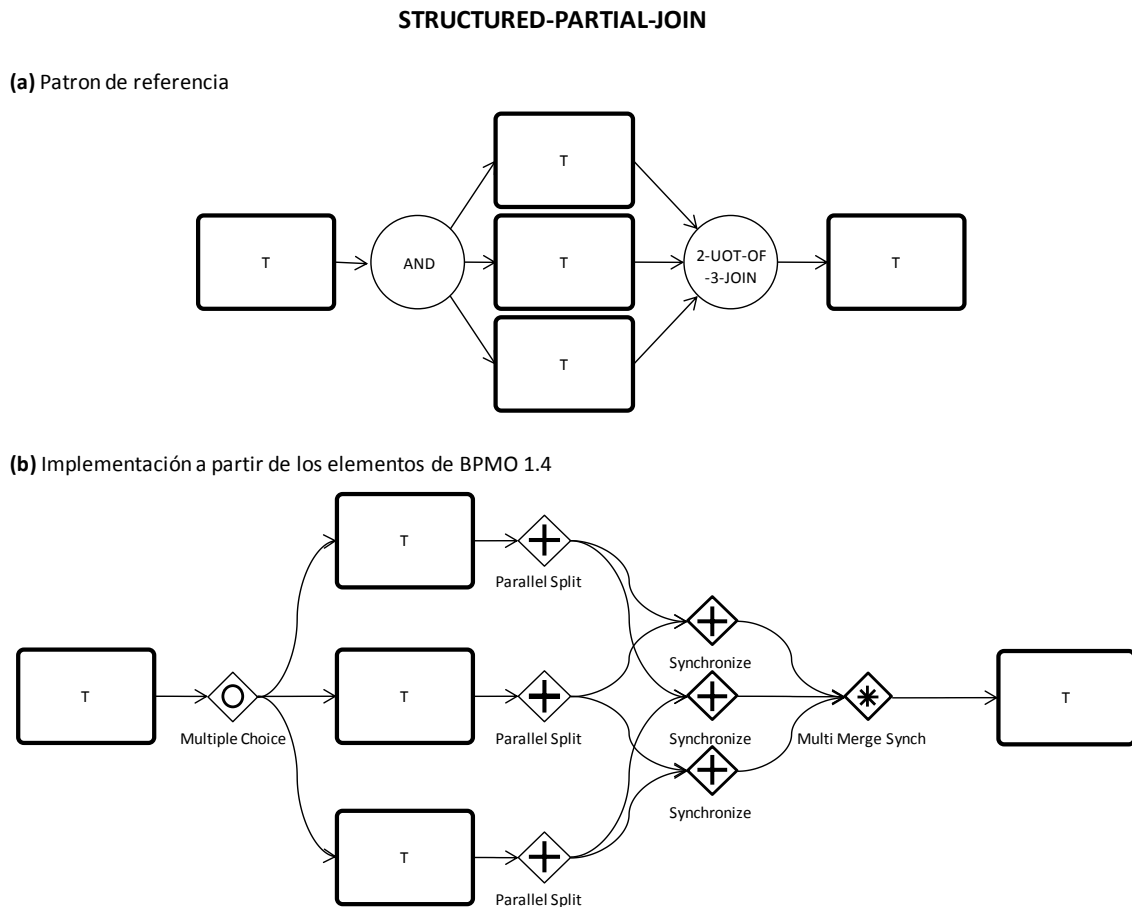


Figura A.3-1: Ejemplo de implementación (Structured-Partial-Join)

A.3.1 Viabilidad de Implementación de los Patrones en BPMO

A continuación en la Figura A.3-2 se muestra un análisis de viabilidad de implementación de los patrones de control de flujo en BPMO de acuerdo a la clasificación presentada en la sección A.2.

PATRONES BASICOS DE FLUJO DE CONTROL		PATRONES DE MULTIPLES INSTANCIAS			
1. Sequence	+	12. Multiple Instances without Synchronization	+		
2. Parallel Split	+	13. Multiple Instances with a Priori Design-Time Knowledge	+		
3. Synchronization	+	14. Multiple Instances with a Priori Run-Time Knowledge	+		
4. Exclusive Choice	+	15. Multiple Instances without a Priori Run-Time Knowledge	+/-		
5. Simple Merge	+	34. Static Partial Join for Multiple Instances	+/-		
		35. Cancelling Partial Join for Multiple Instances	+/-		
		36. Dynamic Partial Join for Multiple Instances	+/-		
PATRONES DE RAMIFICACION Y SINCRONIZACION AVANZADA		PATRONES DE CANCELACION Y TERMINACION FORZADA			
6. Multi-Choice	+	19. Cancel Task	+		
7. Structured Synchronizing Merge	+	20. Cancel Case	+/-		
8. Multi-Merge	+	25. Cancel Region	-		
9. Structured Discriminator	+	26. Cancel Multiple Instance Activity	-		
28. Blocking Discriminator	+/-	27. Complete Multiple Instance Activity	-		
29. Cancelling Discriminator	+/-				
30. Structured Partial Join	+	PATRONES DE ITERACION			
31. Blocking Partial Join	+/-	10. Arbitrary Cycles	+		
32. Cancelling Partial Join	+/-	21. Structured Loop	+		
33. Generalised AND-Join	+	22. Recursion	-		
37. Local Synchronizing Merge	+	PATRONES DE TERMINACION			
38. General Synchronizing Merge	+	11. Implicit Termination	-		
41. Thread Merge	-	43. Explicit Termination	+/-		
42. Thread Split	-	PATRONES DE ACTIVACION			
		23. Transient Trigger	-		
		24. Persistent Trigger	+/-		
PATRONES BASADOS EN ESTADOS		PATRONES DE CANCELACION Y TERMINACION FORZADA			
16. Deferred Choice	+	19. Cancel Task	+		
17. Interleaved Parallel Routing	+	20. Cancel Case	+/-		
18. Milestone	-	25. Cancel Region	-		
39. Critical Section	-	26. Cancel Multiple Instance Activity	-		
40. Interleaved Routing	+/-	27. Complete Multiple Instance Activity	-		
		PATRONES DE ITERACION			
		10. Arbitrary Cycles	+		
		21. Structured Loop	+		
		22. Recursion	-		
		PATRONES DE TERMINACION			
		11. Implicit Termination	-		
		43. Explicit Termination	+/-		
		PATRONES DE ACTIVACION			
		23. Transient Trigger	-		
		24. Persistent Trigger	+/-		
(+) viabilidad de implementacion ALTA		(+/-) viabilidad de implementacion PARCIAL		(-) viabilidad de implementacion NULA	
Cantidad patrones de flujo de control	Viabilidad de implementacion en BPMO				
21	ALTA				
12	PARCIAL				
10	NULA				

Figura A.3-2: Viabilidad de implementación de los patrones en la notación BPMO.

El análisis de viabilidad presentado en la Figura A.3-2 permitió definir un conjunto de patrones que se pueden implementar en el lenguaje BPMO. Dicho conjunto comprende patrones de las sub categorías de: patrones básicos, de ramificación, de sincronización avanzada y de patrones basados en estados. En la próxima sección se hace una descripción detallada de cada uno de los doce patrones.

A.4 Descripción de los Patrones Seleccionados

En esta sección se presentan la selección de patrones compatibles con el lenguaje BPMO, que se utilizaron en la presente propuesta. En total se seleccionó un conjunto de 12 patrones de control de flujo de los 43 definidos por el trabajo de Rusell et al [1], en el cual se puede encontrar una descripción detallada de la funcionalidad de los mismos.

A.4.1 Patrón Secuencia (Sequence)

Este patrón sirve como el bloque fundamental para los modelos BP. Consiste en una secuencia de mínimo tres actividades de tipo tarea “T”, las cuales pueden ser de tipo “*Tarea Manual*”, “*Tarea de Servicio Web*” o “*Tarea de Meta*” de acuerdo al lenguaje BPMO. En este patrón el control de flujo conecta cada tarea precedente con la siguiente, como se puede apreciar por las flechas continuas de la Figura A.4-1.

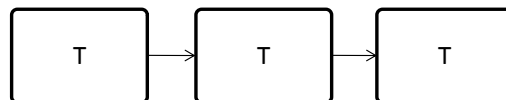


Figura A.4-1: Patrón Secuencia con tres tareas

A.4.2 Patrón División Paralela (Parallel Split)

Este patrón permite dividir el control de flujo en dos o más flujos salientes que ejecutan actividades simultáneamente. En la Figura A.4-2 se presenta un nodo de tipo tarea que una vez ejecutado genera un control de flujo entrante hacia una compuerta “*Parallel Split*”, la cual divide el control de flujo en dos ramas que permiten la posterior ejecución de las dos tareas consiguientes en forma simultánea.

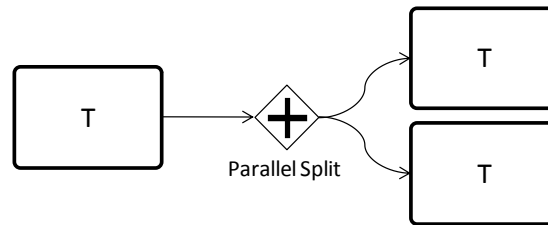


Figura A.4-2: Patrón División Paralela con tres tareas

A.4.3 Patrón Sincronización (Synchronize)

Este patrón permite la convergencia del control de flujo de dos o más ramas en una sola, de tal manera que el control de flujo de la rama saliente ocurra solo cuando el flujo de cada rama entrante converja después de ejecutarse las actividades de cada rama. En la Figura A.4-3 se presentan dos actividades de tipo tarea que terminan su ejecución en momentos diferentes, generando flujos de control asíncronos. Por lo tanto la compuerta “*Synchronize*” espera a que se ejecuten todas las tareas entrantes para generar el control de flujo de salida.

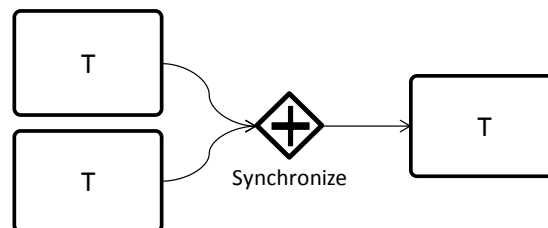


Figura A.4-3: Patrón Sincronización con tres tareas.

A.4.4 Patrón Selección Exclusiva (Exclusive Choice)

Este patrón permite la divergencia del control de flujo de una rama en dos o más ramas, tal que el control de flujo sea direccionado a una actividad específica dependiendo del resultado de bien sea una actividad precedente, los valores de la actividad o algunos datos específicos, los cuales representan la existencia de una decisión para el enrutamiento del control de flujo. En la Figura A.4-4 se presenta una actividad de tipo tarea, que después de ejecutarse genera un control de flujo de

salida hacia una compuerta “*Exclusive Choice*” para luego direccionar el control de flujo por solo una de las ramas salientes, dependiendo de los resultados de la tarea inicial.

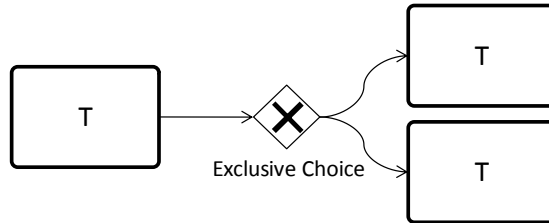


Figura A.4-4: Patrón Selección Exclusiva con tres tareas.

A.4.5 Patrón Combinación Simple (Simple Merge)

Este patrón permite la convergencia asíncrona del control de flujo de dos o más ramas en una sola, de tal manera que el flujo de ejecución se pasa a una sola rama de salida. En la Figura A.4-5 se presentan dos actividades de tipo tarea en las cuales, independiente del momento en que terminen su ejecución, los flujos generados hacia la compuerta “*Simple Merge*” convergen de manera asíncrona ocasionando una ejecución diferente de la tarea posterior para cada uno de los resultados de las tareas iniciales.

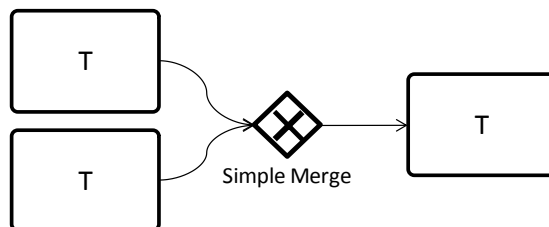


Figura A.4-5. Patrón Combinación Simple con tres tareas.

A.4.6 Patrón Selección Múltiple (Multiple choice)

Este patrón permite la divergencia del control de flujo de una rama en dos o más ramas, tal que el flujo de ejecución se direcciona a cada una de las ramas

salientes de manera síncrona o asíncrona dependiendo de los resultados de la tarea inicial. En la Figura A.4-6 se presenta una actividad de tipo tarea, que después de ejecutarse genera un control de flujo entrante para la compuerta “*Multiple Choice*”, la cual divide de manera síncrona o asíncrona el control de flujo hacia las dos tareas posteriores.

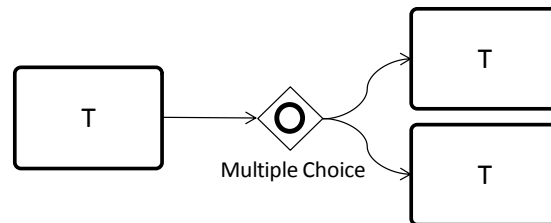


Figura A.4-6. Patrón Selección Múltiple con tres tareas.

A.4.7 Patrón Combinación Múltiple (Multi Merge)

Este patrón permite la convergencia síncrona o asíncrona del control de flujo de dos o más ramas en una sola de tal manera que el flujo de ejecución pasa a una sola rama de salida. En la Figura A.4-7 se muestran dos actividades de tipo tarea, que independiente del momento en que terminen su ejecución cada uno de los flujos generados hacia la compuerta “*Multi Merge*” convergen de manera síncrona o asíncrona ocasionando una ejecución diferente de la tarea posterior para cada uno de los resultados de las tareas iniciales.

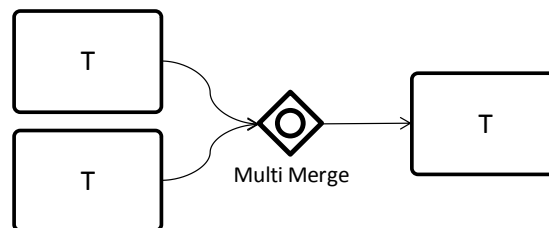


Figura A.4-7. Patrón Combinación Múltiple con tres tareas.

A.4.8 Patrón Selección Diferida (Deferred Choice)

Este patrón permite la divergencia del control de flujo de una rama en dos o más ramas tal que el control de flujo se direcciona a una actividad específica dependiendo del resultado de una actividad precedente. Los valores de la actividad o algunos datos específicos permiten la toma de decisión para el enrutamiento del control de flujo. En la Figura A.4-8 se presenta una actividad de tipo tarea que después de ejecutarse genera un control de flujo de salida hacia una compuerta “*Deferred Choice*” para luego enrutar el control de flujo por una sola de las ramas salientes dependiendo de los resultados de la tarea inicial.

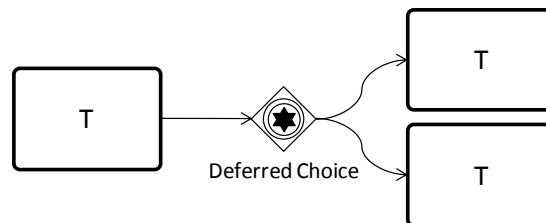


Figura A.4-8. Patrón Selección Diferida con tres tareas.

A.4.9 Patrón Discriminante (Discriminator)

Este patrón permite la convergencia del control de flujo de dos o más ramas en una sola de manera asíncrona de manera tal que el control de flujo se pasa a una sola rama de salida. En la Figura A.4-9 se presentan dos actividades de tipo tarea, que independiente del momento en que terminen su ejecución, cada uno de los flujos generados hacia la compuerta “*Discriminator*” convergen de manera asíncrona. En este caso la compuerta deja pasar el flujo de ejecución generado por la primera tarea ejecutada y solo deja pasar el siguiente una vez que la primera termina su ejecución, ocasionando de esta manera una ejecución diferente para la tarea posterior.

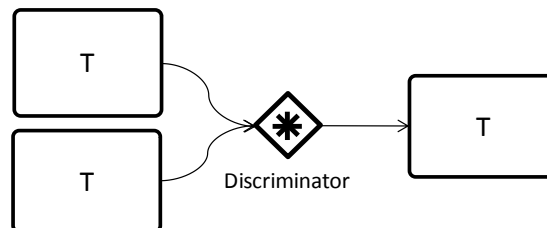


Figura A.4-9. Patrón Discriminante con tres tareas.

A.4.10 Patrón de enrutamiento paralelo intercalado (Interleaved Parallel Routing):

Este patrón permite la divergencia del control de flujo en dos o más ramas, tal que se dirija a una actividad específica dependiendo del resultado de una actividad precedente. En la Figura A.4-10 se presenta una actividad de tipo tarea que una vez ejecutada genera un control de flujo de salida hacia una compuerta “*Interleaved Parallel Routing*” para luego enrutarlo por solo una de las ramas salientes dependiendo de los resultados de la tarea inicial.

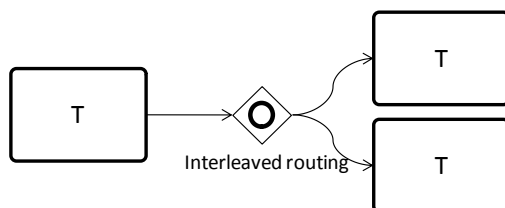


Figura A.4-10: Patrón de enrutamiento paralelo intercalado con 3 tareas.

A.4.11 Patrón de Combinación Múltiple Sincronizada (Multi Merge Synchronize)

Este patrón permite la convergencia del control de flujo de 2 o más ramas en una sola, de tal manera que el control de flujo de la rama saliente ocurra solo cuando el flujo de cada rama entrante converja una vez ejecutadas las actividades de cada rama. En la Figura A.4-11 se presentan 2 actividades de tipo tarea que terminan su ejecución en momentos diferentes generando flujos de control asíncronos; por lo tanto la compuerta “*Multi Merge Synchronization*” espera a que se ejecuten todas las tareas entrantes para generar el control de flujo de salida.

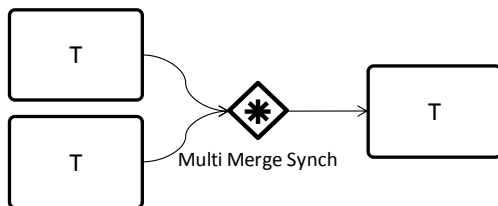


Figura A.4-11. Patrón de Combinación Múltiple Sincronizada con 3 tareas.

A.4.12 Patrón de Instanciación Múltiple (Multiple Instantiation)

Este patrón permite la divergencia del control de flujo de una rama en dos o más, tal que el control de flujo se dirija a una tarea específica dependiendo del resultado de una tarea precedente. En la Figura A.4-12 se presenta una actividad de tipo tarea que después de ejecutarse genera un control de flujo de salida hacia una compuerta “*Multiple Instantiation*” para luego enrutar el control de flujo por solo una de las ramas salientes dependiendo de los resultados de la tarea inicial.

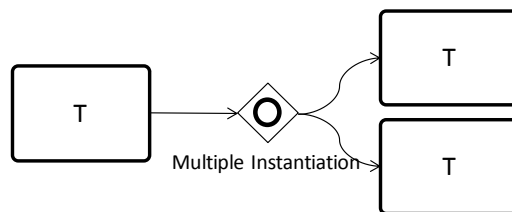


Figura A.4-12. Patrón de Múltiple Instanciación con tres tareas.

A.5 Comportamiento y Funcionalidad de los Patrones

En esta sección se presenta una clasificación de los patrones de acuerdo a su funcionalidad en cuanto a paralelismo y sincronización (AND, XOR, OR y SEQ) y su comportamiento respecto a la convergencia y divergencia de los flujos seleccionados (SPLIT, JOIN y KEEP). Sin embargo, para comprender claramente hay que observar cómo se representan los patrones en el formalismo de grafos de proceso. En la Figura A.5-1 se presentan los doce patrones de control de flujo seleccionados y se puede observar su representación en el lenguaje BPMN y en el formato de grafos de proceso.

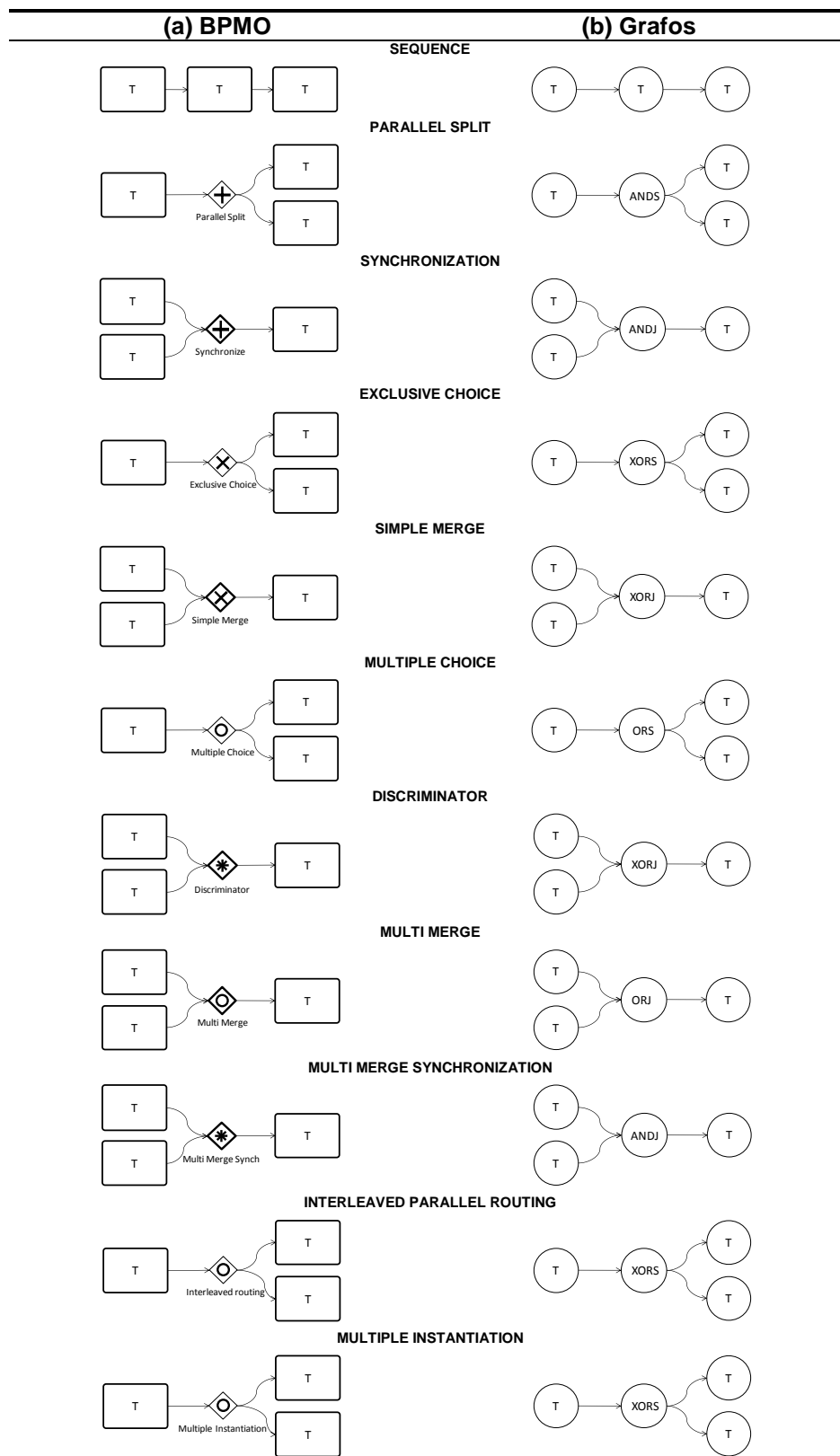


Figura A.5-1: Patrones representados con (a) BPMO y (b) grafos.

A continuación se describen las clasificaciones de los patrones de acuerdo a funcionalidad y comportamiento.

A.5.1 Clasificación de patrones según su funcionalidad

Las categorías definidas para la clasificación de patrones según su funcionalidad son: AND, XOR, OR y SEQ. La primera de ellas denominada AND, hace referencia a un funcionamiento síncrono del control de flujo, es decir que un patrón que sincronice el control de flujo puede ser clasificado como tipo AND. La segunda categoría es denominada XOR y hace referencia a un funcionamiento asíncrono del control de flujo, es decir que un patrón que permita generar asincronismo en el control de flujo, puede ser clasificado como tipo XOR. La tercera categoría es denominada OR y su funcionamiento consiste en que de acuerdo a una serie de condiciones establecidas, puede permitir la sincronía o asincronismo del control de flujo. Finalmente, La cuarta categoría es denominada SEQ, y hace referencia a un funcionamiento secuencial del control de flujo en el cual no se presenta sincronía ni asincronismo.

A.5.2 Clasificación de patrones según su comportamiento

Las categorías definidas para la clasificación de patrones según su comportamiento son SPLIT, JOIN y KEEP. La primera de estas tres categorías denominada SPLIT hace referencia a los patrones que presentan un comportamiento de divergencia en el control de flujo en dos o más ramas, en contraste con la categoría ya mencionada, La segunda categoría denominada JOIN hace referencia a los patrones que presentan un comportamiento de convergencia en el control de flujo de dos o más ramas, Finalmente, La tercera categoría denominada KEEP, hace referencia a los patrones que no realizan divergencia ni convergencia del control de flujo, es decir, aquellos que mantienen la coherencia del control de flujo.

De acuerdo a las anteriores clasificaciones se generó la Tabla A.5-1a la cual resume la clasificación de los patrones básicos definidos para BPMO según su funcionalidad y comportamiento.

Nombre del Patrón	Funcionalidad				Comportamiento		
	AND	OR	XOR	SEQ	SPLIT	JOIN	KEEP
Sequence				+			+
Parallel Split	+				+		
Synchronize	+					+	
Simple Merge			+			+	
Multi Merge Synch	+					+	
Discriminator			+			+	
Multi Merge		+				+	
Deferred Choice			+		+		
Multiple Instantiation			+		+		
Interleaved Routing			+		+		
Multiple Choice		+			+		
Exclusive Choice			+		+		

Tabla A.5-1. Clasificación en funcionalidad y comportamiento de los patrones.

En esta tabla el patrón “Sequence” tiene una funcionalidad SEQ y un comportamiento KEEP lo que significa que en este patrón el control de flujo es secuencial y coherente. Seguidamente el patrón “Parallel Split”, el cual tiene una funcionalidad AND y un comportamiento SPLIT lo que significa que sincroniza y diverge el control de flujo. Luego, el patrón “Synchronize” presenta una funcionalidad AND y un comportamiento JOIN, lo que significa que sincroniza y converge el control de flujo, así como el patrón “Multi Merge Synch”. A continuación, se aborda el patrón “Simple Merge” que posee una funcionalidad XOR y un comportamiento JOIN, lo que significa que el control de flujo es asíncrono y convergente así como en el patrón “Discriminator”. Seguidamente, es presentado el patrón denominado “Multi Merge”, cuya funcionalidad es OR y su comportamiento JOIN, esto significa que el control de flujo es convergente y puede ser síncrono o asíncrono dependiendo del conjunto de condicionales que posea. A continuación, el patrón “Deferred Choice” presenta una funcionalidad XOR y un comportamiento SPLIT, lo que significa que el control de flujo es asíncrono y diverge, así como los patrones “Multiple Instantiation”, “Interleaved Routing” y “Exclusive Choice”. Finalmente, se aborda el patrón “Multiple Choice” el cual tiene una funcionalidad OR y un comportamiento SPLIT, esto significa que el control de flujo es divergente y puede ser síncrono o asíncrono dependiendo del conjunto de condicionales que posea.

A.6 Relaciones entre los patrones de control de flujo

En esta sección se definen las relaciones de especialización y composición existentes entre los patrones de control de flujo básicos definidos para BPMO, las cuales se describen a continuación.

A.6.1 Relación de especialización entre patrones

Una relación de *especialización* entre patrones existe donde uno es una forma más restringida de otro. Por ejemplo el patrón “Multiple Choice” es una forma más especializada del patrón “Parallel Split. En otras palabras el patrón “Parallel Split” presenta una funcionalidad más general correspondiente a un AND y un comportamiento SPLIT lo cual indica que el control de flujo diverge de manera síncrona.

La Figura A.6-1, muestra una secuencia de imágenes que describen la funcionalidad AND y el comportamiento SPLIT del patrón “Parallel Split”, en la cual se puede apreciar que el control de flujo diverge de manera síncrona a través de cada una de sus ramas. En especial se puede apreciar que la secuencia 4 de la imagen (remarcada con un cuadro azul), corresponde con la funcionalidad síncrona del patrón “Multiple Choice” (recuadro rojo), Pero este a su vez puede generar un control de flujo síncrono o asíncrono (recuadros verdes) (debido a que presenta una funcionalidad de tipo OR,) dependiendo de las condiciones del control de flujo de entrada. Por consiguiente, la funcionalidad síncrona o asíncrona del patrón “Multiple Choice” lo acerca más a una relación de exclusión, definiendo así su relación de especialización respecto al patrón “Parallel Split”.

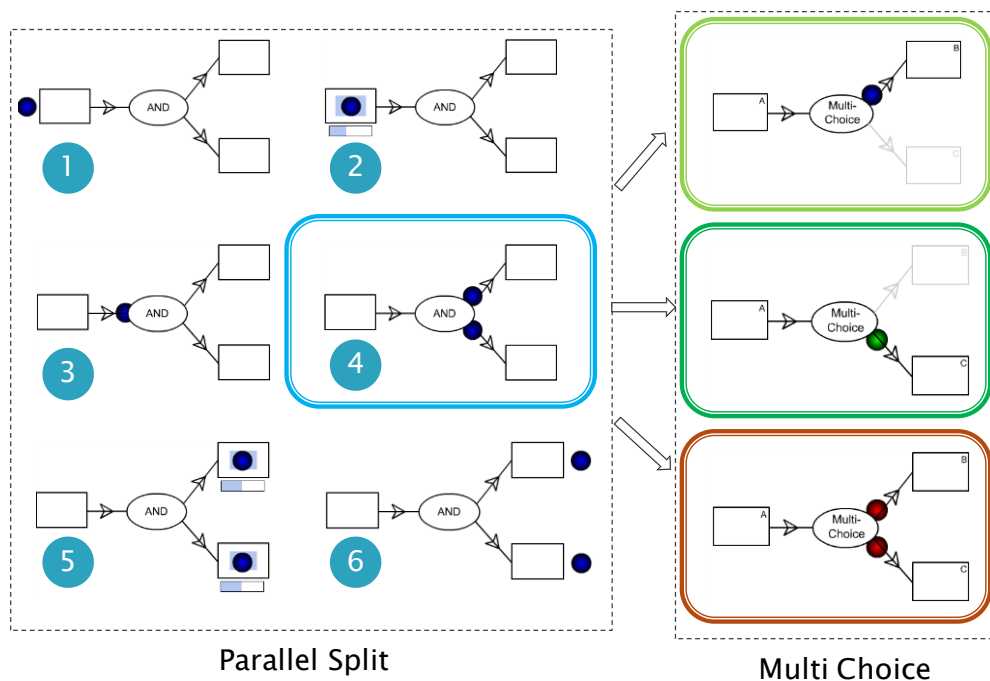


Figura A.6-1: Relación de especialización entre los patrones Parallel Split y Multiple Choice.

A continuación en la Tabla A.6-1 se presentan todas las relaciones de especialización entre los patrones de control de flujo de referencia definidos en el trabajo de Russell et al[1]. En este trabajo los patrones se clasifican en dos grupos. El primero de ellos (parte izquierda - Tabla A.6-1) contiene a los patrones denominados patrones especializados cuya funcionalidad es excluyente como se mencionó en la sección anterior. El segundo grupo (parte derecha - Tabla A.6-1), contiene patrones denominados patrones generales, y a diferencia del primer grupo, su funcionalidad es generalizada.

Nr	Specialized Pattern Name	General Pattern
3	Synchronization	31. Blocking Partial Join
		32. Cancelling Partial Join (only where N=M)
		33. Generalized AND-Join
		38. General Synchronizing Merge
4	Exclusive Choice	6. Multi-Choice
5	Simple Merge	8. Multi-Merge
6	Multi-choice	2. Parallel Split
7	Structured Synchronizing Merge	37. Acyclic Synchronizing Merge
9	Structured Discriminator	28. Blocking Discriminator
		30. Structured Partial Join
13	MIs with a priori D/T Knowledge	14. MIs with a priori R/T Knowledge
14	MIs with a priori R/T Knowledge	15. MIs without a priori R/T Knowledge
19	Cancel Activity	25. Cancel Region
20	Cancel Case	25. Cancel Region
26	Cancel Multiple Instance Activity	19. Cancel Activity
27	Complete MI Activity	26. Cancel MI Activity
28	Blocking Discriminator	31. Blocking Partial Join
29	Cancelling Discriminator	32. Cancelling Partial Join
30	Structured Partial Join	31. Blocking Partial Join
34	Static Partial Join for MIs	36. Dynamic Partial Join for MIs
37	Acyclic Synchronizing Merge	38. General Synchronizing Merge
40	Interleaved Routing	17. Interleaved Parallel Routing

Tabla A.6-1: Relaciones de especialización entre patrones de control de flujo.
Tomada del trabajo de Russell et al [1]

A.6.2 Relación de composición entre patrones

En algunos casos, es posible que un patrón pueda ser expresado a través de una combinación de dos (o más) patrones. Por ejemplo, el patrón “Multiple Choice” que tiene una funcionalidad OR y un comportamiento SPLIT (control de flujo síncrono o asíncrono y divergente), se puede expresar mediante una combinación de funcionalidades del patrón “Parallel Split” (control de flujo síncrono y divergente) y el patrón “Exclusive Choice” (control de flujo asíncrono y divergente). En consecuencia, este tipo de relación se denomina relación de *composición* entre patrones, debido a que la funcionalidad del control de flujo de un patrón es el resultado de la composición (superposición de las funcionalidades) de dos o más patrones.

En la Figura A.6-2, se presenta una secuencia de imágenes que describe de manera detallada el ejemplo mencionado anteriormente para el patrón “Multiple Choice”. En la parte inicial de esta secuencia (recuadro de contorno azul al lado

izquierdo de la Figura A.6-2), se observa una representación de la funcionalidad divergente y síncrona del control de flujo del patrón “Parallel Split”, que combinada con la funcionalidad divergente y asíncrona del patrón “Exclusive Choice” (recuadros de contorno azul en la parte central de la Figura A.6-2), permiten expresar la funcionalidad síncrona o asíncrona y divergente correspondiente al patrón “Multiple Choice” (recuadros al lado derecho de la Figura A.6-2).

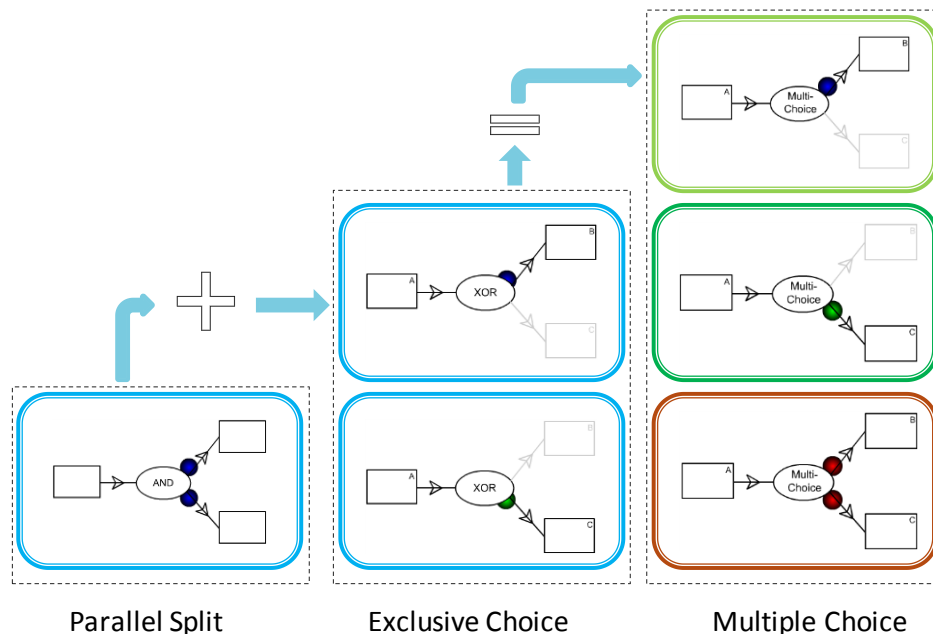


Figura A.6-2. Relación de composición entre los patrones Parallel Split, Exclusive Choice y Multiple Choice.

La Tabla A.6-2, presenta todas las relaciones de composición entre los patrones de control de flujo de referencia definidos en el trabajo de Russell et al [1], en el cual los patrones se clasifican en dos grupos. El primero de ellos (parte izquierda - Tabla A.6-2) se denomina patrones elementales, y su principal característica, consiste en que su funcionalidad no se puede expresar como la combinación de las funcionalidades de los demás patrones. El segundo grupo (parte derecha - Tabla A.6-2), se denomina patrones compuestos, y a diferencia del primer grupo, los patrones que integran este grupo se caracterizan por tener una funcionalidad que puede ser expresada como una combinación de las funcionalidades de los patrones elementales.

Nr	Pattern Name	Composed From
6	Multi-Choice	2. Parallel Split in conjunction with 4. Exclusive Choice
10	Arbitrary Cycles ⁹	4. Exclusive Choice together with 5. Simple Merge
12	MI without Synchronization ⁴	6. Multi-Choice together with 8. Multi-Merge
21	Structured Loop ⁴	4. Exclusive Choice together with 8. Multi-Merge
29	Cancelling Discriminator	25. Cancel Region together with 28. Blocking Discriminator
32	Cancelling Partial Join	25. Cancel Region together with 31. Blocking Partial Join
35	Cancelling Partial Join for MIs	34. Static Partial Join for MIs together with 26. Cancel MI Activity

Tabla A.6-2. Relaciones de composición entre patrones de control de flujo
Tomada del trabajo de Russell et al [1]

Más generalmente en la Figura A.6-3 se pueden observar las relaciones de especialización (líneas continuas) y las relaciones de composición (líneas punteadas) existentes entre los 43 patrones de referencia descritos en [1]. Además se puede notar que solo existen relaciones de especialización o composición entre 32 de los 43 patrones de referencia.

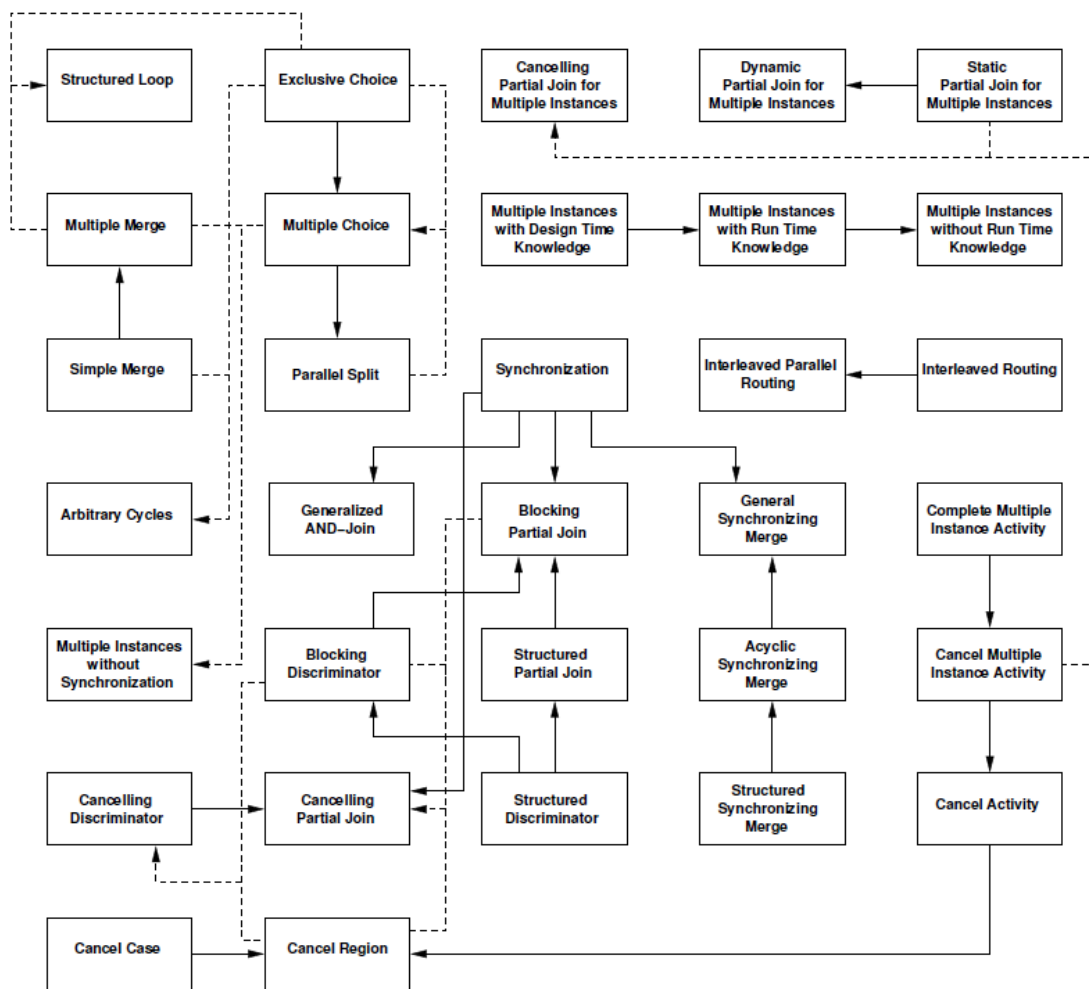


Figura A.6-3. Relaciones de especialización y composición entre patrones de control de flujo

Tomada del trabajo de Russell et al [1]

A partir de las relaciones entre patrones descritas en este documento, se realizó una abstracción, en la cual se tienen en cuenta únicamente las relaciones aplicables entre los 12 patrones básicos de control de flujo implementados en BPMO. La Figura A.6-4 muestra la abstracción de relaciones mencionada a partir de un diagrama que contiene 5 recuadros para representar los patrones básicos implementados en BPMO y líneas entre ellos que representan las relaciones de especialización y composición entre ellos. En la figura se pueden evidenciar; la relación de composición entre los patrones “Exclusive Choice” y “Parallel Split” (denotada por las flechas punteadas) los cuales componen el patrón “Multiple Choice”; y por otra parte,

las relaciones de especialización presentes (denotadas por flechas continuas), corresponden a la especialización del patrón “Parallel Split” que respectivamente es el patrón “Multiple Choice”; la especialización del patrón “Multiple Choice” que respectivamente es el patrón “Exclusive Choice”; y la especialización del patrón “Multi Merge” que respectivamente es el patrón “Simple Merge”.

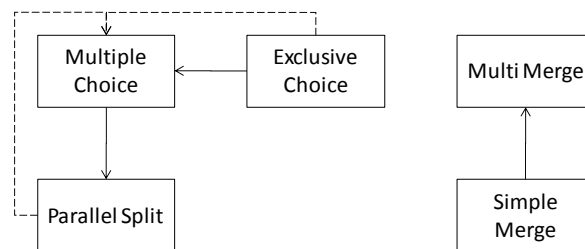


Figura A.6-4. Relaciones de especialización y composición para los patrones básicos de BPMO.

A.7 Ontología de Patrones de Control de Flujo

A partir de la abstracción de las relaciones entre patrones básicos realizada en la sección A.6 y la clasificación de patrones básicos según funcionalidad y comportamiento en esta sección se generó la ontología para patrones básicos de control de flujo descritos en BPMO. Para lo cual, inicialmente se identificaron los principales conceptos relacionados al dominio del control de flujo de los procesos de negocio con el fin de establecer los súper conceptos de los cuales heredarían atributos los conceptos pertenecientes a la ontología de patrones de control de flujo.

De esta manera, el primer concepto padre para la ontología de patrones de control de flujo se denominó “controlflowpattern” y hace referencia a cualquier tipo de patrón de control de flujo. Por lo tanto, cada uno de los 12 patrones básicos de control de flujo, se representó por medio de una instancia del concepto padre “controlflowpattern”. Esta instancia contiene los siguientes atributos:

- **hasFunction:** corresponde a una instancia del concepto “function” la cual puede tomar los valores AND, XOR, OR o SEQ.
- **hasBehaviour:** corresponde a una instancia del concepto “behaviour” la cual puede tomar los valores SPLIT, JOIN o KEEP.
- **isSpecializedOf:** corresponde a una instancia del concepto “controlflowpattern” la cual puede tomar los valores “Sequence”, “Parallel Split”, “Synchronize”, “Simple Merge”, “Multi Merge Synch”, “Discriminator”, “Multi Merge”, “Deferred Choice”, “Multiple Instantiation”, “Interleaved Routing”, “Multiple Choice” y “Exclusive Choice”.
- **composeTo:** corresponde a una instancia del concepto “controlflowpattern” la cual puede tomar los valores “Sequence”, “Parallel Split”, “Synchronize”, “Simple Merge”, “Multi Merge Synch”, “Discriminator”, “Multi Merge”, “Deferred Choice”, “Multiple Instantiation”, “Interleaved Routing”, “Multiple Choice” y “Exclusive Choice”.

Por otra parte, el segundo concepto padre de la ontología de patrones de control de flujo se denominó “function”, el cual hace referencia al tipo de funcionalidad que puede tener cada patrón (AND, XOR, OR, SEQ). De esta forma, cada tipo de funcionalidad, es a su vez representada por una instancia del concepto padre “function”. El tercer concepto padre de la ontología de patrones de control de flujo, fue denominado “behaviour”, el cual hace referencia al tipo de comportamiento que puede tener cada patrón (SPLIT, JOIN, KEEP). Cada tipo de comportamiento es a su vez representado por una instancia del concepto padre “behaviour”.

La Figura A.7-1 muestra la representación gráfica de la ontología de patrones básicos de control de flujo para BPMO propuesta en este trabajo; en la cual los nodos representan conceptos o instancias y las aristas las relaciones de súper conceptos y subconceptos existentes entre los conceptos de la ontología.

Para la implementación de la ontología de patrones de control de flujo, se utilizó un entorno EMF (Eclipse Media Framework) denominado WSMT (Web Service Modeling Toolkit), el cual provee funcionalidades de edición para la creación de


```

unicauca#location hasValue "Popayan-Cauca-Colombia"
unicauca#title hasValue "Universidad del Cauca"
subject hasValue "Semantic Business Process Management"
title hasValue "BPMOONER Ontology"
version hasValue "1.0.0"
author hasValue "https://sites.google.com/site/bpmooner/"
git#title hasValue "Grupo de Ingenieria Telematica"
country hasValue "Colombia"
endNonFunctionalProperties

importsOntology
_ "http://ip-super.org/ontologies/process/bpmo/v2.0.1#bpmo"

concept processmodelrepository subConceptOf _ "http://ip-
super.org/ontologies/process/bpmo/v2.0.1#BusinessResource"
nonFunctionalProperties
description hasValue "location for process model storage"
endNonFunctionalProperties

concept processmodel subConceptOf _ "http://ip-
super.org/ontologies/process/bpmo/v2.0.1#Process"
nonFunctionalProperties
description hasValue "An organized arrangement of systems concepts and principles
that portray the behavior of a system through time"
endNonFunctionalProperties

concept bpmoonercontrolflowpattern
nonFunctionalProperties
description hasValue "open standards directly observable categorising recurring
problems and solutions in a particular domain"
endNonFunctionalProperties
hasBehaviour ofType bpmoonercbehaviour
hasFunction ofType bpmoonercfunction
isSpecializedOf ofType bpmoonerccontrolflowpattern
composeTo ofType bpmoonerccontrolflowpattern

concept bpmoonersplit
nonFunctionalProperties
description hasValue "split the control flow in two or more flows"
endNonFunctionalProperties

concept bpmoonercjoin
nonFunctionalProperties
description hasValue "join in a common flow two or more incoming flows"
endNonFunctionalProperties

concept bpmoonerckeep
nonFunctionalProperties
description hasValue "control flow goes on"
endNonFunctionalProperties

concept bpmoonercseq
nonFunctionalProperties
description hasValue "sequential flow of proceso de negocio elements"
endNonFunctionalProperties

concept bpmoonercand
nonFunctionalProperties
description hasValue "results are obtained simultaneously"
endNonFunctionalProperties

concept bpmoonercor
nonFunctionalProperties
description hasValue "results are obtained in simultaneously or partially"
endNonFunctionalProperties

concept bpmoonercxor
nonFunctionalProperties
description hasValue "results are obtained exclusively partial"
endNonFunctionalProperties

```

```

concept bpmoonersequence subConceptOf { bpmoonercontrolflowpattern, bpmoonerseq}
  nonFunctionalProperties
    description hasValue "It is used to construct a series of consecutive activities
which execute in turn one after the other"
  endNonFunctionalProperties

concept bpmoonerdeferredchoice subConceptOf { bpmoonercontrolflowpattern, bpmoonersplit,
bpmoonerxor}
  nonFunctionalProperties
    description hasValue "Provides the ability to defer the moment of choice in a
process"
  endNonFunctionalProperties

concept bpmoonerexclusivechoice subConceptOf { bpmoonercontrolflowpattern, bpmoonersplit,
bpmoonerxor}
  nonFunctionalProperties
    description hasValue "Allows the thread of control to be directed to a specific
activity depending on the outcome of a preceding activity"
  endNonFunctionalProperties

concept bpmoonermultiplechoice subConceptOf { bpmoonercontrolflowpattern, bpmooneror,
bpmoonersplit}
  nonFunctionalProperties
    description hasValue "Pattern provides the ability for the thread of execution to be
diverged into several concurrent threads on a selective basis"
  endNonFunctionalProperties

concept bpmoonermultipleinstantiation subConceptOf { bpmoonercontrolflowpattern,
bpmoonersplit, bpmoonerxor}
  nonFunctionalProperties
    description hasValue "Offers the possibility of relaxing the strict ordering that a
process usually imposes over a set of activities"
  endNonFunctionalProperties

concept bpmoonersynchronisation subConceptOf { bpmoonercontrolflowpattern, bpmoonerand,
bpmoonerjoin}
  nonFunctionalProperties
    description hasValue "Provides a means of reconverging the execution threads of two
or more parallel branches"
  endNonFunctionalProperties

concept bpmoonersimplemerge subConceptOf { bpmoonercontrolflowpattern, bpmoonerxor,
bpmoonerjoin}
  nonFunctionalProperties
    description hasValue "Provides a means of merging two or more distinct branches
without synchronizing them"
  endNonFunctionalProperties

concept bpmoonerparallelsplit subConceptOf { bpmoonercontrolflowpattern, bpmoonerand,
bpmoonersplit}
  nonFunctionalProperties
    description hasValue "Allows a single thread of execution to be split into two or
more branches which can execute activities concurrently"
  endNonFunctionalProperties

concept bpmoonerdiscriminator subConceptOf { bpmoonercontrolflowpattern, bpmoonerxor,
bpmoonerjoin}
  nonFunctionalProperties
    description hasValue "Provides a mechanism for progressing the execution of a
process once the first of a series of concurrent activities has completed"
  endNonFunctionalProperties

concept bpmoonerinterleavedparallelrouting subConceptOf { bpmoonercontrolflowpattern,
bpmoonersplit, bpmoonerxor}
  nonFunctionalProperties
    description hasValue "Offers the possibility of relaxing the strict ordering that a
process usually imposes over a set of activities"
  endNonFunctionalProperties

```

```

concept bpmoonermultimerge subConceptOf { bpmoonercontrolflowpattern, bpmoonerjoin,
bpmooneror}
  nonFunctionalProperties
    description hasValue "Provides a means of merging distinct branches in a process
into a single branch"
  endNonFunctionalProperties

concept bpmoonermultiplemergesynchronise subConceptOf { bpmoonercontrolflowpattern,
bpmoonerand, bpmoonerjoin}
  nonFunctionalProperties
    description hasValue "Provides a means of merging the branches resulting from a
specific Multi-Choice or OR-split construct earlier in a proceso de negocio process into a
single branch"
  endNonFunctionalProperties

concept bpmoonerbehaviour
  nonFunctionalProperties
    description hasValue "can split or merge or keep the control flow"
  endNonFunctionalProperties

concept bpmoonerfunction
  nonFunctionalProperties
    description hasValue "can be simultaneous (AND) or partial and simultaneous (OR) or
exclusive (XOR)"
  endNonFunctionalProperties

instance SEQ memberOf { bpmoonerseq, bpmoonerfunction}

instance AND memberOf { bpmoonerand, bpmoonerfunction}

instance OR memberOf { bpmooneror, bpmoonerfunction}

instance XOR memberOf { bpmoonerxor, bpmoonerfunction}

instance JOIN memberOf { bpmoonerjoin, bpmoonerbehaviour}

instance SPLIT memberOf { bpmoonersplit, bpmoonerbehaviour}

instance KEEP memberOf { bpmoonerkeep, bpmoonerbehaviour}

instance Sequence memberOf { bpmoonersequence, bpmoonercontrolflowpattern}
  hasBehaviour hasValue KEEP
  hasFunction hasValue SEQ

instance DeferredChoice memberOf { bpmoonerdeferredchoice, bpmoonercontrolflowpattern}
  hasBehaviour hasValue SPLIT
  hasFunction hasValue XOR

instance ExclusiveChoice memberOf { bpmoonerexclusivechoice, bpmoonercontrolflowpattern}
  hasBehaviour hasValue SPLIT
  hasFunction hasValue XOR
  isSpecializedOf hasValue {MultipleChoice, ParallelSplit }
  composeTo hasValue MultipleChoice

instance MultipleChoice memberOf { bpmoonermultiplechoice, bpmoonercontrolflowpattern}
  hasBehaviour hasValue SPLIT
  hasFunction hasValue OR
  isSpecializedOf hasValue ParallelSplit

instance MultipleInstantiation memberOf { bpmoonermultipleinstantiation,
bpmoonercontrolflowpattern}
  hasBehaviour hasValue SPLIT
  hasFunction hasValue XOR

instance Synchronisation memberOf { bpmoonersynchronisation, bpmoonercontrolflowpattern}
  hasBehaviour hasValue JOIN
  hasFunction hasValue AND

instance SimpleMerge memberOf { bpmoonersimplemerge, bpmoonercontrolflowpattern}
  hasBehaviour hasValue JOIN

```

```

hasFunction hasValue XOR
isSpecializedOf hasValue MultiMerge

instance ParallelSplit memberOf { bpmoonerparallelsplit, bpmoonercontrolflowpattern}
hasBehaviour hasValue SPLIT
hasFunction hasValue AND
composeTo hasValue MultipleChoice

instance Discriminator memberOf { bpmoonerdiscriminator, bpmoonercontrolflowpattern}
hasBehaviour hasValue JOIN
hasFunction hasValue XOR

instance InterleavedParallelRouting memberOf { bpmoonerinterleavedparallelrouting,
bpmoonercontrolflowpattern}
hasBehaviour hasValue SPLIT
hasFunction hasValue XOR

instance MultiMerge memberOf { bpmoonermultimerge, bpmoonercontrolflowpattern}
hasBehaviour hasValue JOIN
hasFunction hasValue OR

instance MultiMergeSynchronise memberOf { bpmoonermultiplemergesynchronise,
bpmoonercontrolflowpattern}
hasBehaviour hasValue JOIN
hasFunction hasValue AND

```

Tabla A.7-1: Ontología de Patrones de Control de Flujo en formato WSMML

Anexo B

B Lenguajes Semánticos y Ontologías de Dominio

B.1 Introducción

En esta sección se destaca la importancia de la semántica para el enriquecimiento de los procesos de negocio de un operador de telecomunicaciones. En primer lugar se definen los lenguajes semánticos para la descripción de procesos de negocio y ontologías existentes. Posteriormente se presenta el estudio y selección de las ontologías de dominio de telecomunicaciones más apropiadas para el enriquecimiento de las tareas que conforman los procesos de negocio. Por último se realiza una descripción detallada de la adaptación de dichas ontologías dentro del presente proyecto, mediante la definición de un mecanismo de enriquecimiento semántico de tareas.

B.2 Lenguajes de Descripción de Ontologías

El crecimiento descontrolado en la generación y compartición del conocimiento en internet, ha propiciado la especificación formal de la información a través de modelos computacionalmente comprensibles como las ontologías. En consecuencia,

los lenguajes primitivos para el intercambio de información tales como KIF, Ontolingua, Frame Logic, entre otros [3], han sido reemplazados por lenguajes de descripción más avanzados como RDF (Resource Description Framework), DAML (DARPA Agent Markup Language), OIL (Ontology Inference Layer) y OWL (Web Ontology Language), los cuales han permitido explotar las características de la web de una forma más eficiente, y han facilitado los procesos de razonamiento, inferencia y evaluación automática sobre las descripciones de los recursos.

La gestión de servicios y procesos de negocio no ha sido ajena a esta tendencia respecto al manejo de la información, en los últimos años las organizaciones han promovido la utilización de ontologías para el enriquecimiento semántico de los documentos descriptores de sus recursos (servicios y procesos de negocio), como una medida para reducir la ambigüedad y facilitar el entendimiento común de los objetivos del negocio en todos los niveles de la empresa.

Es por esto que en la actualidad existen numerosos lenguajes estándares de descripción de ontologías empleados tanto para el manejo de la información disponible en la web como para la gestión semántica de procesos de negocio. A continuación se presenta una breve descripción de los lenguajes más conocidos y ampliamente utilizados para la descripción de ontologías.

B.2.1 RDF (Resource Description Framework)

RDF es un lenguaje de propósito general desarrollado por el W3C¹ (World Wide Web Consortium) para representar información de los recursos que pueden ser identificados en la web. RDF proporciona un marco común para expresar esta información de manera que pueda ser intercambiada entre aplicaciones sin perder su significado o sentido[4], mediante la especificación de enunciados simples acerca de los recursos empleando propiedades y valores (tripleas RDF). Sin embargo, este lenguaje no permite precisar cuándo las descripciones de los recursos se refieren a un tipo o clase específica de un dominio [4]. Por esta razón se hizo necesario utilizar un mecanismo para definir vocabularios especializados en los enunciados, el cual se

¹ Disponible en la dirección www.w3.org/Consortium/

conoce como Lenguaje de Descripción de Vocabularios en RDF (RDFS por sus siglas en inglés) [5].

B.2.2 RDFS (RDF Schema)

RDFS es una extensión semántica de RDF, que define un mecanismo para describir las propiedades y las clases de los recursos RDF. RDFS provee terminologías definidas sobre RDF que permiten el modelado de objetos con una semántica definida, gracias a la cual es posible especificar las propiedades aplicables a las clases de objetos[5].

RDFS proporciona un conjunto de términos que permiten construir las expresiones o sentencias acerca de los recursos, representar sus propiedades y establecer, mediante relaciones taxonómicas, jerarquías de generalización entre las clases. Además, mediante este lenguaje es posible instanciar objetos a partir de las clases usando propiedades, y establecer algunas restricciones sobre dichas propiedades [6].

Por otra parte, este lenguaje permite construir ontologías simples sobre las cuales se puede realizar consultas y razonamiento automático, sin embargo no es lo suficientemente expresivo para ontologías complejas, ya que solo permite realizar inferencia sobre la herencia de propiedades y no posee la capacidad para generar axiomas [7] [8] [9, 10].

Para solucionar los inconvenientes presentados con RDF y RDFS se desarrollaron otros lenguajes como OIL, DAML-ONT (DARPA Agent Markup Language - Lenguaje de Marcas de Agentes de DARPA) y DAML+OIL. OIL permite mejor interoperabilidad semántica entre recursos web que RDF, y emplea un modelo para la representación del conocimiento basado en lógica descriptiva (axiomas y reglas) y en sistemas basados en marcos (taxonomía de clases y atributos). Las principales carencias de este lenguaje son la falta de expresividad para declarar axiomas y para soportar dominios concretos. Por su parte, DAML-ONT fue desarrollado como una extensión de XML y RDF para extender el nivel de expresividad de RDFS. DAML+OIL corresponde a una unión de los anteriores lenguajes y las características de OIL, alejándose un poco del modelo basado en

marcos y potenciando la lógica descriptiva [11]. A partir de esta unión se definió el lenguaje OWL, con el propósito de reunir todas las ventajas de DAML+OIL y estandarizar el conocimiento extendido de RDF.

B.2.3 OWL (Web Ontology Language)

OWL es un lenguaje de marcado semántico creado por la W3C para publicar y compartir ontologías en la web. En relación a sus predecesores, OWL añade más vocabulario para describir propiedades y clases; permite establecer relaciones adicionales entre las clases (por ejemplo de disyunción); y atribuye ciertas propiedades a las relaciones, como por ejemplo de cardinalidad, simetría, transitividad o relaciones inversas. OWL incluye toda la capacidad expresiva de RDFS y la extiende con la posibilidad de utilizar expresiones lógicas [12].

Si bien estas características hacen de OWL un lenguaje poderoso en expresión, también generan dificultades en cuanto a la complejidad y capacidad de decisión computacional [13, 14], lo cual limita la capacidad de los razonadores para inferir nuevo conocimiento a partir de sentencias dadas. Por esta razón, se crearon tres variantes o sub-lenguajes de OWL que proveen expresividad incremental con propiedades computacionales diferentes: OWL Lite, OWL DL y OWL Full [13]. En la actualidad, OWL es el lenguaje de representación de ontologías más utilizado en el contexto de la web semántica, principalmente porque se constituyó en un estándar avalado por el W3C.

B.2.4 WSML (Web Service Modeling Language)

WSML es un lenguaje de modelado que define un mecanismo formal para describir los elementos definidos en WSMO (*Web Service Modelling Ontology*), la cual es una ontología formal desarrollada por un conjunto de empresas coordinadas por la institución de investigación de empresas digitales (DERI por sus siglas en inglés). La ontología WSMO cual proporciona un modelo conceptual para la descripción de diversos aspectos relacionados con servicios web semánticos e

identifica cuatro elementos de alto nivel que agrupan los conceptos principales que tienen que ser descritos para poder definir un servicio web semántico: las *Ontologías* que proporcionan la terminología usada por otros elementos de WSMO; los *Servicios Web* que representan entidades computacionales capaces de proporcionar acceso a los servicios; las *Metas* que describen aspectos relacionados con los deseos de los usuarios respecto a la funcionalidad solicitada; y los *Mediadores* cuyo objetivo es el manejo automático de problemas de interoperabilidad entre diferentes elementos de WSMO [15].

Si bien WSMO proporciona el modelo conceptual para describir los anteriores elementos (entre los que se encuentran las ontologías), dicho lenguaje carece de una formalización real de los conceptos para que puedan ser computacionalmente procesables. Es por esto que la W3C recomienda utilizar WSML como el lenguaje formal para la escritura, almacenamiento y comunicación de los conceptos definidos en WSMO [16].

WSML está basado en diferentes lógicas formales denominadas: Lógica de Descripción (Description Logics), Lógica de Primer Orden (First-Order Logic) y Programación Lógica (Logic Programming), las cuales se utilizan para el modelado de servicios web semánticos. De acuerdo a estas lógicas formales se definieron cinco variantes de WSML: WSML-Core, WSML-DL, WSML-Flight, WSML-Rule y WSML-Full, cada una con distinto nivel de expresividad lógica Figura B.2-1. De estas cinco variantes, la más adecuada para describir ontologías es WSML-Flight ya que es la variante más expresiva y combina tanto la lógica de descripción como la de primer orden con una complejidad computacional manejable [16].

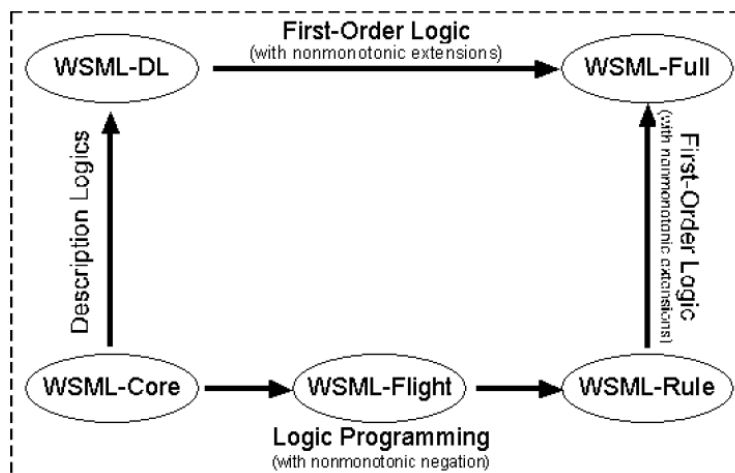


Figura B.2-1: Variantes de WSML

Una de las ventajas más importantes que presenta WSML es que se especifica en una sintaxis legible, y adicionalmente tiene una sintaxis XML y RDF para el intercambio de datos entre servicios, lo que le brinda interoperabilidad con otras aplicaciones basadas en RDF. Además, WSML discrimina claramente entre sintaxis conceptual y sintaxis de expresión lógica, permitiendo modelar distintivamente los elementos conceptuales de WSMO (servicio web, ontologías, metas y mediadores), de manera independiente de la descripción de restricciones adicionales y axiomas [17].

B.2.5 Selección del lenguaje semántico de descripción de ontologías.

Tanto las ontologías como los servicios necesitan ser especificados en lenguajes formales con el fin de permitir su procesamiento automatizado. El lenguaje que recomienda la W3C para la descripción de ontologías es OWL, sin embargo, a pesar de que este lenguaje posee un gran potencial en cuanto a expresividad lógica, presenta limitaciones tanto a nivel conceptual como en la definición de algunas de sus propiedades formales [16, 18]. Por esta razón se hizo necesaria la creación de un lenguaje conceptual que permitiera resolver el problema de la integración de aplicaciones, mediante la definición de una tecnología coherente y con buenas propiedades computacionales para describir servicios web semánticos, el cual se conoce como WSMO.

Uno de los elementos principales de WSMO son las ontologías, sin embargo este modelo conceptual por si solo es insuficiente para automatizar su procesamiento, y por ende requiere de un lenguaje formal de descripción como WSML, el cual proporciona una expresividad lógica razonable y una semántica formal bien definida que facilitan el razonamiento lógico sobre las ontologías y demás elementos de WSMO [19].

Por lo tanto en la presente propuesta se definió el lenguaje WSML como el lenguaje semántico para describir las ontologías utilizadas dentro del proyecto, ya se presenta como una alternativa menos compleja y por lo tanto más comprensible en relación con OWL. Además, WSML es uno de los lenguajes más avanzados dentro de la industria de los servicios web semánticos y posee numerosas herramientas disponibles en la web que facilitan su utilización, tales como WSMO Studio, WSML Rule Reasoner, WSMO4J, WSMT (Web Service Modeling Toolkit), entre otras.

B.3 Entorno de Modelado de Procesos de Negocio BPMO

Para llevar a cabo el modelado de los modelos de procesos en la notación BPMO se utilizó el modelador BPMO en su versión 1.4, el cual corresponde a un “plugin” del WSMO Studio [20]; éste último es un EMF (Eclipse Modeling Framework) para desarrollo de proyectos de software semántico. El Modelador BPMO (Figura B.3-1) provee una gran variedad de funcionalidades, entre las cuales están:

- **WSMO editor plugin** es usado para crear las descripciones de todos los elementos WSML: ontologías, servicios, objetivos y mediadores.
- **Editor de Coreografía** es usado para crear las descripciones centrales de las coreografías de los servicios en WSMO. Permite la conexión remota a ontologías, objetivos y repositorios de servicios desde su interfaz de usuario.

- **Validador WSML** es usado para detectar problemas e inconsistencias dentro de las descripciones en los elementos WSML.
- **Razonador WSML-Flight** permite verificar que el rendimiento de las ontologías creadas, sea satisfactorio. Permite el uso de varios componentes de descubrimiento para alcanzar objetivos basados en el descubrimiento de servicios.
- **Editor SAWSDL**, que provee las funcionalidades para describir la información de mapeado entre las descripciones existentes entre los documentos WSDL y WSML. Contiene varios plugin de terceros que pueden ser usados desde su interface: el adaptador IRS-III, el Infrawebs Axiom Editor y el WSML Visualizer.
- **Editor BPMO** provee la funcionalidad de crear modelos de procesos de negocio semánticos de acuerdo a Business Process Modelling Ontology (BPMO).

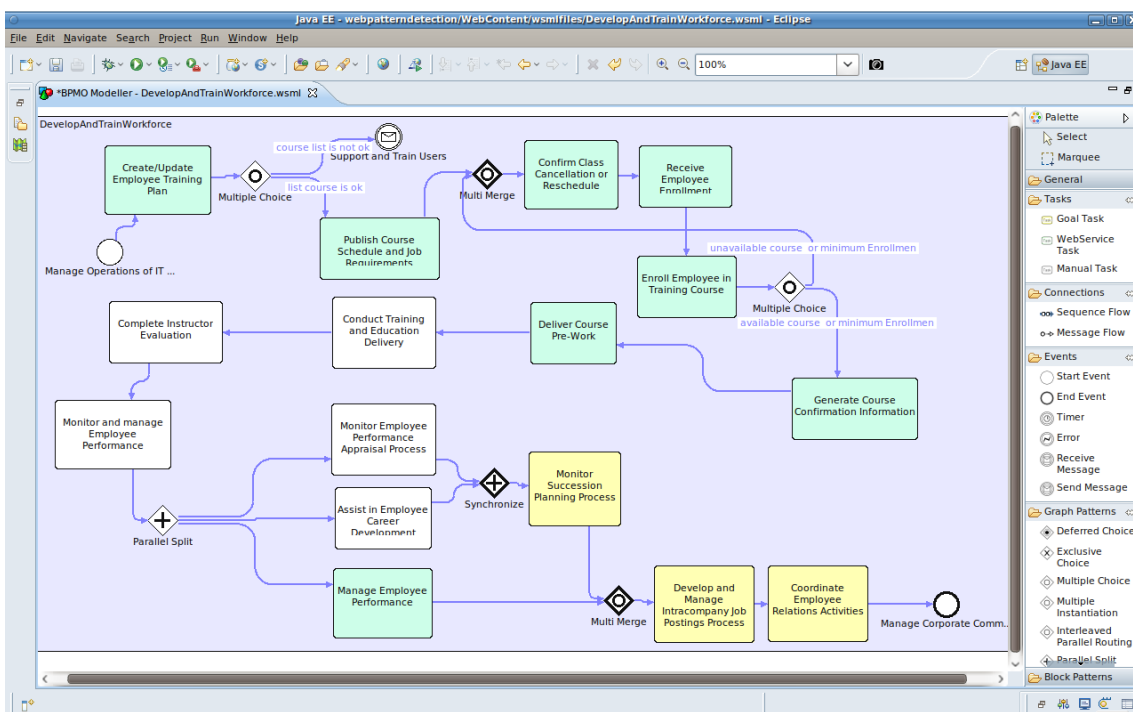


Figura B.3-1: Modelador WSMO Studio.

En la Figura B.3-2 se encuentra la representación gráfica de un BP Model del ejemplo llamado “*Develop and Train Workforce*”, modelado a partir de BPMO Modeler 1.4 descrito anteriormente. En esta es posible observar los eventos representados por elementos de contorno circular, las tareas representadas por elementos de contorno cuadrado con las esquinas redondeadas y las compuertas representadas por elementos de contorno en forma de rombo, así como los conectores de flujo de control representados por flechas que indican el orden y la secuencia en la cual se ejecutan las actividades.

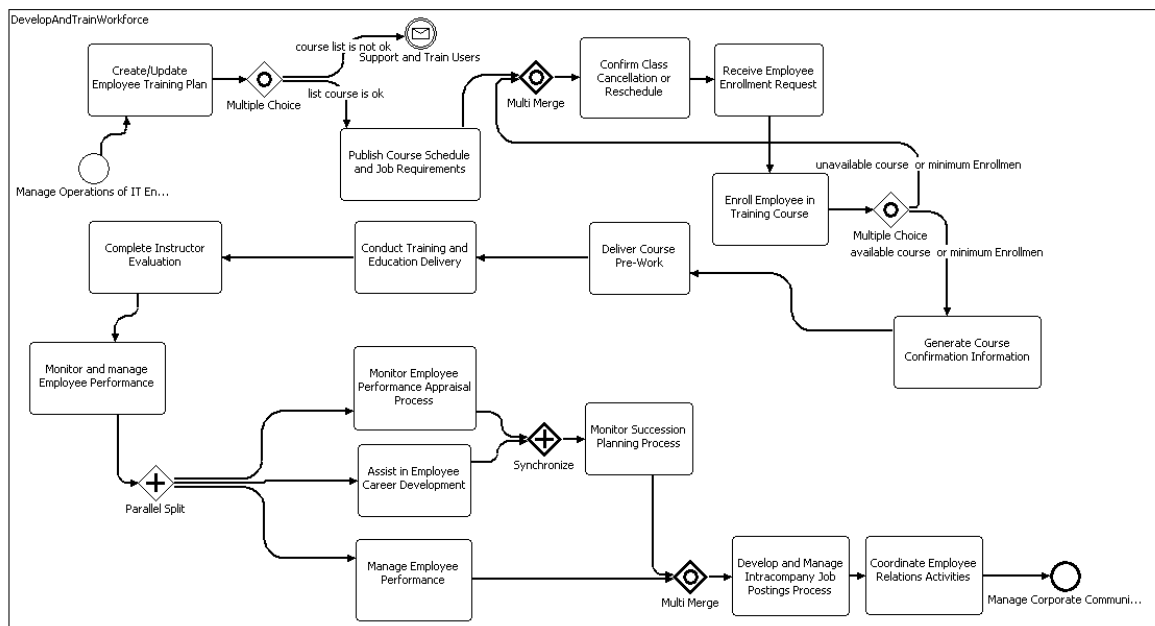


Figura 2

Figura B.3-2: Representación gráfica del ejemplo “*Develop and Train Workforce*” descrito en WSML a partir del BPMO Modeler 1.4

A continuación, en la Tabla B.3-1 se presenta el contenido del archivo del procesos de negocio de ejemplo llamado “*Develop and train workforce.wsmf*”, el cual cuenta con los siguientes elementos:

- **namespace:** contiene una URL asociada a la ontología utilizada para la descripción del BP Model, que para el marco de este trabajo de grado corresponde a bpmo-1-4-20080109 y otra URL asociada al EMF para desarrollo de proyectos de software semántico llamado WSMO Studio, el cual fue utilizado a lo largo del presente trabajo de grado como herramienta de modelado.

- **ontology:** corresponde al identificador de la instancia que se hace de la ontología para describir el BP Model.
- **importsOntology:** importación de la ontología bpmo-1-4-20080109 a partir de la cual se describe el BP Model.
- **instance:** instancias de los conceptos de la ontología bpmo-1-4-20080109 indicados por la palabra reservada “memberOf”.

Las demás líneas corresponden a los atributos de cada instancia de concepto de la ontología bpmo-1-4-20080109 y su valor puede ser una instancia de otro concepto o un tipo de dato los cuales son indicados por la palabra reservada “hasValue”. Adicionalmente es posible agregar propiedades no funcionales indicadas por la palabra reservada “nonFunctionalProperties”.

```

namespace { _"http://www.ip-super.org/ontologies/BPMO/bpmo-1-4-20080109#"
'
  wsmostudio _"http://www.wsmostudio.org#" }

ontology instanceOntology_1265864406355_2062259045
  nonFunctionalProperties
    wsmostudio#version hasValue "0.7.3"
  endNonFunctionalProperties

  importsOntology
    _"http://www.ip-super.org/ontologies/BPMO/bpmo-1-4-20080109#"

instance Task_1265869249595_203985252 memberOf Task
  hasName hasValue "Assist in Employee Career Development"
  hasDescription hasValue "to enhance an employee's skills to optimize performance
and reinforce the employee's ability to take on broader responsibilities; or changes in
your organization's functions"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue EmployeePerformance
  sendsData hasValue EmployeeCareerDevelopment

instance Process_1265864442709_1778458876 memberOf Process
  hasName hasValue "DevelopAndTrainWorkforce"
  hasDescription hasValue "Workforce development has emerged to describe a relatively
wide range of activities, policies, and programs."
  hasWorkflow hasValue Workflow_1265869249596_1816462921

instance Workflow_1265869249596_1816462921 memberOf Workflow
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasWorkflowElement hasValue StartEvent_1265869249592_148649460

instance StartEvent_1265869249592_148649460 memberOf StartEvent
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasName hasValue "Manage Operations of IT Environment"

instance ControlflowConnector_1265869249599_1182940149 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue Task_1265869249595_203985252
  hasTarget hasValue Synchronisation_1265869249595_347179985

instance Synchronisation_1265869249595_347179985 memberOf Synchronisation

```

```

hasHomeProcess hasValue Process_1265864442709_1778458876
hasIncomingConnector hasValue {ControlflowConnector_1265869249599_1182940149,
ControlflowConnector_1265869249599_894559872 }

instance ControlflowConnector_1265869249599_894559872 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue Task_1265869249595_689927293
hasTarget hasValue Synchronisation_1265869249595_347179985

instance Task_1265869249595_689927293 memberOf Task
hasName hasValue "Monitor Employee Performance Appraisal Process"
hasDescription hasValue "Most companies have a formal performance appraisal system
in which employee job performance is rated on a regular basis, usually once a year. A
good performance appraisal system can greatly benefit an organization."
hasHomeProcess hasValue Process_1265864442709_1778458876
receivesData hasValue EmployeePerformance
sendsData hasValue EmployeePerformanceAppraisalProcess

instance EmployeePerformance memberOf DataEntity
hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#PerformanceMeasure"

instance EmployeePerformanceAppraisalProcess memberOf DataEntity
hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#PerformanceMeasure"

instance ControlflowConnector_1265869249598_2012925208 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue Task_1265869249594_2086344716
hasTarget hasValue Task_1265869249594_1783693746

instance Task_1265869249594_2086344716 memberOf Task
hasName hasValue "Complete Instructor Evaluation"
hasDescription hasValue "This evaluation instrument is used to improve the teaching
effectiveness of the faculty member and the learning process of the employee"
hasHomeProcess hasValue Process_1265864442709_1778458876
receivesData hasValue TrainingandEducationDelivery
sendsData hasValue InstructorEvaluation

instance TrainingandEducationDelivery memberOf DataEntity
hasContent hasValue "http://ip-super.org/ontologies/domain/YATOSP/sid/v1.3.0#File"

instance InstructorEvaluation memberOf DataEntity
hasContent hasValue "http://ip-super.org/ontologies/domain/YATOSP/sid/v1.3.0#File"

instance Task_1265869249594_1783693746 memberOf Task
hasName hasValue "Monitor and manage Employee Performance"
hasDescription hasValue "Employee performance management is a program that
corporate management uses to evaluate the current performance of employees and to help
improve the performance in the future. Employee performance management is a never-ending
process to help the company and its workers"
hasHomeProcess hasValue Process_1265864442709_1778458876
receivesData hasValue InstructorEvaluation
sendsData hasValue EmployeePerformance

instance EmployeePerformance memberOf DataEntity
hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#PerformanceMeasure"

instance ControlflowConnector_1265869249598_1328882639 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue StartEvent_1265869249592_148649460
hasTarget hasValue WebServiceTask_1265869249592_436141531

instance WebServiceTask_1265869249592_436141531 memberOf WebServiceTask
hasName hasValue "Create and Update Employee Training Plan"
hasDescription hasValue "Employee training programs provide focused training to a
number of employees either on a one-time or recurring basis"
hasHomeProcess hasValue Process_1265864442709_1778458876
receivesData hasValue CalendarOperationsIT

```

```

sendsData hasValue EmployeeTrainingPlan

instance CalendarOperationsIT memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#StandardWorkCalendar"

instance EmployeeTrainingPlan memberOf DataEntity
  hasContent hasValue "http://ip-super.org/ontologies/domain/YATOSP/sid/v1.3.0#Plan"

instance ControlflowConnector_1267034668884_1630364385 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue ParallelSplit_1267034668883_624727885
  hasTarget hasValue Task_1265869249595_689927293

instance ParallelSplit_1267034668883_624727885 memberOf ParallelSplit
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasOutgoingConnector hasValue {ControlflowConnector_1267034668884_1466658587,
ControlflowConnector_1267034668884_1630364385,
ControlflowConnector_1267034668885_239734334 }

instance ControlflowConnector_1267034668884_1466658587 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue ParallelSplit_1267034668883_624727885
  hasTarget hasValue Task_1265869249595_203985252

instance ControlflowConnector_1267034668885_239734334 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue ParallelSplit_1267034668883_624727885
  hasTarget hasValue WebServiceTask_1267034668884_689721832
instance WebServiceTask_1267034668884_689721832 memberOf WebServiceTask
  hasName hasValue "Manage Employee Performance"
  hasDescription hasValue "performance management includes activities to ensure that
goals are consistently being met in an effective and efficient manner. Performance
management can focus on performance of the organization, a department, processes to
build a product or service, employees, etc"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue EmployeePerformance
  sendsData hasValue ManageEmployee

instance ManageEmployee memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#ResourcePerformance"

instance ConditionalflowConnector_1267821735879_407988819 memberOf
ConditionalflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue MultipleChoice_1267821735878_76911386
  hasTarget hasValue MultiMerge_1267821735878_1980064553
  hasCondition hasValue Condition_1270858614574_1375648916

instance MultipleChoice_1267821735878_76911386 memberOf MultipleChoice
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasOutgoingConnector hasValue {ConditionalflowConnector_1267821735879_1663076259,
ConditionalflowConnector_1267821735879_407988819 }

instance ConditionalflowConnector_1267821735879_1663076259 memberOf
ConditionalflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue MultipleChoice_1267821735878_76911386
  hasTarget hasValue WebServiceTask_1265869249594_401151975
  hasCondition hasValue Condition_1270858614574_325208820

instance WebServiceTask_1265869249594_401151975 memberOf WebServiceTask
  hasName hasValue "Generate Course Confirmation Information"
  hasDescription hasValue "confirmation that the course is to be made to current
employees"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue TrainingCourseList
  sendsData hasValue CourseConfirmationInformation

```

```

instance TrainingCourseList memberOf DataEntity
  hasContent hasValue "http://ip-super.org/ontologies/domain/YATOSP/sid/v1.3.0#File"

instance CourseConfirmationInformation memberOf DataEntity
  hasContent hasValue "http://ip-super.org/ontologies/domain/YATOSP/sid/v1.3.0#File"

instance Condition_1270858614574_325208820 memberOf Condition
  hasExpression hasValue "available course or minimum Enrollmen"

instance MultiMerge_1267821735878_1980064553 memberOf MutiMerge
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasIncomingConnector hasValue {ConditionalflowConnector_1267821735879_407988819,
ControlflowConnector_1267821735879_1494926135 }

instance ControlflowConnector_1267821735879_1494926135 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue WebServiceTask_1265869249592_631884850
  hasTarget hasValue MultiMerge_1267821735878_1980064553

instance WebServiceTask_1265869249592_631884850 memberOf WebServiceTask
  hasName hasValue "Publish Course Schedule and Job Requirements"
  hasDescription hasValue "Before filling a job, it is wise to analyze the job to
determine what's required for optimal performance"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue EmployeeTrainingPlan
  sendsData hasValue CourseSchedule

instance CourseSchedule memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#ProjectSchedule"

instance Condition_1270858614574_1375648916 memberOf Condition
  hasExpression hasValue "unavailable course or minimum Enrollmen"

instance ControlflowConnector_1265869249597_543964881 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue MultiMerge_1265869249595_1754541056
  hasTarget hasValue GoalTask_1265869249595_1478228053

instance MultiMerge_1265869249595_1754541056 memberOf MutiMerge
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasIncomingConnector hasValue {ControlflowConnector_1265869249597_980210718,
ControlflowConnector_1267034668884_1739067018 }

instance ControlflowConnector_1265869249597_980210718 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue GoalTask_1265869249595_1225592672
  hasTarget hasValue MultiMerge_1265869249595_1754541056

instance GoalTask_1265869249595_1225592672 memberOf GoalTask
  hasName hasValue "Monitor Succession Planning Process"
  hasDescription hasValue "A business often represents a lifetime of work and vision.
However, despite almost three-quarters of business owners wanting to transfer control or
exit ownership within the next decade, barely a third have a formal succession plan in
place."
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue {EmployeeCareerDevelopment,
EmployeePerformanceAppraisalProcess }
  sendsData hasValue SuccessionPlanningProcess

instance EmployeeCareerDevelopment memberOf DataEntity
  hasContent hasValue "http://ip-super.org/ontologies/domain/YATOSP/sid/v1.3.0#File"

instance SuccessionPlanningProcess memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#AgreementTermOrCondition"

instance ControlflowConnector_1267034668884_1739067018 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue WebServiceTask_1267034668884_689721832

```

```

hasTarget hasValue MultiMerge_1265869249595_1754541056

instance GoalTask_1265869249595_1478228053 memberOf GoalTask
  hasName hasValue "Develop and Manage Intracompany Job Postings Process"
  hasDescription hasValue "the process of transferring a corporation's technology
between international units of that company for ultimate transfer to their external
customers"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue {ManageEmployee, SuccessionPlanningProcess }
  sendsData hasValue IntracompanyJobPostingsProcess

instance IntracompanyJobPostingsProcess memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#ProjectElementRelationship"

instance ControlflowConnector_1267821735878_217417251 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue WebServiceTask_1267821735878_2121098092
  hasTarget hasValue WebServiceTask_1267821735878_917240235

instance WebServiceTask_1267821735878_2121098092 memberOf WebServiceTask
  hasName hasValue "Confirm Class Cancellation or Reschedule"
  hasDescription hasValue "Notices will appear on the day the class is cancelled
along with any instructions from the instructor"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue CourseSchedule
  sendsData hasValue CourseReschedule

instance CourseReschedule memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#ProjectSchedule"

instance WebServiceTask_1267821735878_917240235 memberOf WebServiceTask
  hasName hasValue "Receive Employee Enrollment Request"
  hasDescription hasValue "Receive Employee Enrollment Request for training"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue CourseReschedule
  sendsData hasValue EmployeeEnrollmentRequest

instance EmployeeEnrollmentRequest memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#RequestResponseRelationship"

instance ControlflowConnector_1267821735879_277026841 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue MultiMerge_1267821735878_1980064553
  hasTarget hasValue WebServiceTask_1267821735878_2121098092

instance ControlflowConnector_1267821735879_384954600 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue WebServiceTask_1267821735878_1666146599
  hasTarget hasValue MultipleChoice_1267821735878_76911386

instance WebServiceTask_1267821735878_1666146599 memberOf WebServiceTask
  hasName hasValue "Enroll Employee in Training Course"
  hasDescription hasValue "means an agreed number of lessons within an agreed period
of time"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue EmployeeEnrollmentRequest
  sendsData hasValue TrainingCourseList

instance ConditionalflowConnector_1265869249598_502874903 memberOf
ConditionalflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue MultipleChoice_1265869249593_574487768
  hasTarget hasValue SendMessageEvent_1265869249593_1339298540
  hasCondition hasValue Condition_1270858614574_1267366404

instance MultipleChoice_1265869249593_574487768 memberOf MultipleChoice
  hasHomeProcess hasValue Process_1265864442709_1778458876

```

```

hasOutgoingConnector hasValue {ConditionalflowConnector_1265869249598_1934137231,
ConditionalflowConnector_1265869249598_502874903 }

instance ConditionalflowConnector_1265869249598_1934137231 memberOf
ConditionalflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue MultipleChoice_1265869249593_574487768
  hasTarget hasValue WebServiceTask_1265869249592_631884850
  hasCondition hasValue Condition_1270858614574_2045606226

instance Condition_1270858614574_2045606226 memberOf Condition
  hasExpression hasValue "list course is ok"

instance SendMessageEvent_1265869249593_1339298540 memberOf SendMessageEvent
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasName hasValue "Support and Train Users"

instance Condition_1270858614574_1267366404 memberOf Condition
  hasExpression hasValue "course list is not ok"

instance ControlflowConnector_1265869249598_7538651 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue WebServiceTask_1265869249594_401151975
  hasTarget hasValue WebServiceTask_1265869249594_1588923813

instance WebServiceTask_1265869249594_1588923813 memberOf WebServiceTask
  hasName hasValue "Deliver Course Pre-Work"
  hasDescription hasValue "number of meetings that discussed the method of course
work"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue CourseConfirmationInformation
  sendsData hasValue CoursePreWork

instance CoursePreWork memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#SimpleWorksProgram"

instance ControlflowConnector_1267034668885_1763999780 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue Task_1265869249594_1783693746
  hasTarget hasValue ParallelSplit_1267034668883_624727885

instance ControlflowConnector_1265869249597_645941398 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue GoalTask_1265869249595_1478228053
  hasTarget hasValue GoalTask_1265869249595_1415749288

instance GoalTask_1265869249595_1415749288 memberOf GoalTask
  hasName hasValue "Coordinate Employee Relations Activities"
  hasDescription hasValue "For motivating employees you can do few things which sound
very ordinary but the employee will feel great. No need to spend large amount, just
small things."
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue IntracompanyJobPostingsProcess
  sendsData hasValue EmployeeRelationsActivities

instance EmployeeRelationsActivities memberOf DataEntity
  hasContent hasValue "http://ip-
super.org/ontologies/domain/YATOSP/sid/v1.3.0#CompletedActivityStatus"

instance ControlflowConnector_1265869249599_171017254 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1265864442709_1778458876
  hasSource hasValue WebServiceTask_1265869249594_1588923813
  hasTarget hasValue Task_1265869249594_382424142

instance Task_1265869249594_382424142 memberOf Task
  hasName hasValue "Conduct Training and Education Delivery"
  hasDescription hasValue "Conduct Training and Education Delivery for employee"
  hasHomeProcess hasValue Process_1265864442709_1778458876
  receivesData hasValue CoursePreWork

```



```

sendsData hasValue TrainingandEducationDelivery

instance ControlflowConnector_1265869249597_770698751 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue WebServiceTask_1265869249592_436141531
hasTarget hasValue MultipleChoice_1265869249593_574487768

instance ControlflowConnector_1265869249599_358926220 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue Task_1265869249594_382424142
hasTarget hasValue Task_1265869249594_2086344716

instance ControlflowConnector_1265869249597_1354130490 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue Synchronisation_1265869249595_347179985
hasTarget hasValue GoalTask_1265869249595_1225592672

instance ControlflowConnector_1265869249596_1492521017 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue GoalTask_1265869249595_1415749288
hasTarget hasValue EndEvent_1265869249596_1831754518

instance EndEvent_1265869249596_1831754518 memberOf EndEvent
hasHomeProcess hasValue Process_1265864442709_1778458876
hasName hasValue "Manage Corporate Communications"

instance ControlflowConnector_1267821735879_663389840 memberOf ControlflowConnector
hasHomeProcess hasValue Process_1265864442709_1778458876
hasSource hasValue WebServiceTask_1267821735878_917240235
hasTarget hasValue WebServiceTask_1267821735878_1666146599

```

Tabla B.3-1: Contenido del archivo del Modelo BP del ejemplo “*Develop and Train Workforce.wsmf*”

B.4 Ontologías del Dominio de las Telecomunicaciones

Una de las claves para lograr la interoperabilidad de los sistemas o aplicaciones es contar con un lenguaje estandarizado que permita el entendimiento común de los conceptos. En esta medida surge la necesidad de encontrar un modelo que permita obtener una visión general de los conceptos más utilizados por los operadores de telecomunicaciones, y que además sea computacionalmente comprensible (es decir que pueda ser interpretado y procesado por una aplicación o sistema software); estas características están proporcionadas por las ontologías de dominio de telecomunicaciones, las cuales son necesarias para llevar a cabo el propósito del presente proyecto, puesto que constituyen la herramienta para enriquecer semánticamente los parámetros que definen las funcionalidades de las tareas de los

procesos de negocio (descritas en términos de sus entradas, salidas e identificadores).

Dentro de los trabajos estudiados se encontraron varios proyectos relacionados con ontologías de dominio de telecomunicaciones [21, 22] [23] [24] [25] [26] [27, 28]; sin embargo gran parte de las ontologías utilizadas en estos se limitan a modelar el conocimiento de un sector muy específico o de una empresa particular, por lo tanto su aplicación dentro del presente proyecto no es viable en la medida en que no favorece el carácter genérico y la escalabilidad de la solución propuesta.

No obstante, uno de los proyectos importantes que se encontraron es un marco de referencia semántico para el sector de las telecomunicaciones denominado YATOSP (Yet Another Telecommunication Ontologies, Process And Services Framework), el cual busca proporcionar un puente entre la arquitectura del proyecto SUPER y un conjunto de ontologías basadas en la iniciativa NGOSS del TMF denominadas SeTOM, SSID y STAM, las cuales definen procesos de negocio semánticos en el dominio de telecomunicaciones mediante la extracción de información común entre las compañías de este sector [29, 30]. A pesar de que existe una buena documentación en cuanto a la descripción de la arquitectura de YATOSP, este trabajo carece de ejemplos de ejecución que permitan verificar la validez de sus planteamientos y restringe el acceso a algunos documentos de gran relevancia para su posible implementación.

El marco de referencia YATOSP sirvió para centrar la atención en la iniciativa NGOSS, especialmente en el conjunto de modelos propuestos para la gestión de los procesos de negocio, la reingeniería de procesos y la integración de aplicaciones empresariales propias del sector de las telecomunicaciones: el Mapa de Operaciones de Telecomunicaciones Mejorado (eTOM, enhanced Telecom Operations Map), el Modelo de Información/Datos Compartidos (SID, Shared Information/Data Model), el Mapa de Aplicaciones de Telecomunicaciones (TAM, Telecom Applications Map) y la Arquitectura Neutral de Tecnologías (TNA, Technology Neutral Architecture)[31].

Teniendo como referencia estas propuestas fue posible establecer una relación de correspondencia entre la forma en que se describen las tareas de un proceso de negocio y la información contenida en las ontologías. De esta manera, el identificador

o nombre de una tarea de un proceso de negocio se puede modelar a través del estándar eTOM, mientras que las entradas y salidas se pueden modelar a través de las entidades que conforman el modelo SID. Para entender mejor estas relaciones, a continuación se presenta una descripción de las principales características de estos dos modelos (eTOM y SID) y de las ontologías desarrolladas a partir de ellos (SeTOM y SSID).

B.4.1 eTOM (enhanced Telecom Operations Map).

eTOM es un marco de referencia de procesos de negocio, que proporciona un modelo para la categorización de todas las actividades involucradas en la operación de un proveedor de servicios de telecomunicaciones. Provee una visión común (sustentada en una terminología común) de la industria de las telecomunicaciones en cuanto a procesos e información, la cual facilita y soporta las relaciones operador-a-operador, operador-a-cliente y operador-a-socio/proveedor respondiendo con esto a la necesidad que representa el rápido despliegue de servicios y el manejo de problemas que exige el entorno competitivo actual [32].

De esta manera, eTOM ordena de una forma jerárquica y coherente los conceptos que constituyen el conocimiento relativo a los procesos, subprocesos y actividades que se llevan a cabo dentro de un proveedor de servicios de telecomunicaciones. Estos procesos pueden clasificarse en cuatro niveles, cada uno de los cuales detalla a su vez procesos más específicos. El nivel 0 corresponde a las tres grandes áreas que abarca el modelo eTOM: Estrategia-Infraestructura-Producto, Operaciones y Gestión Empresarial (Figura B.4-1).

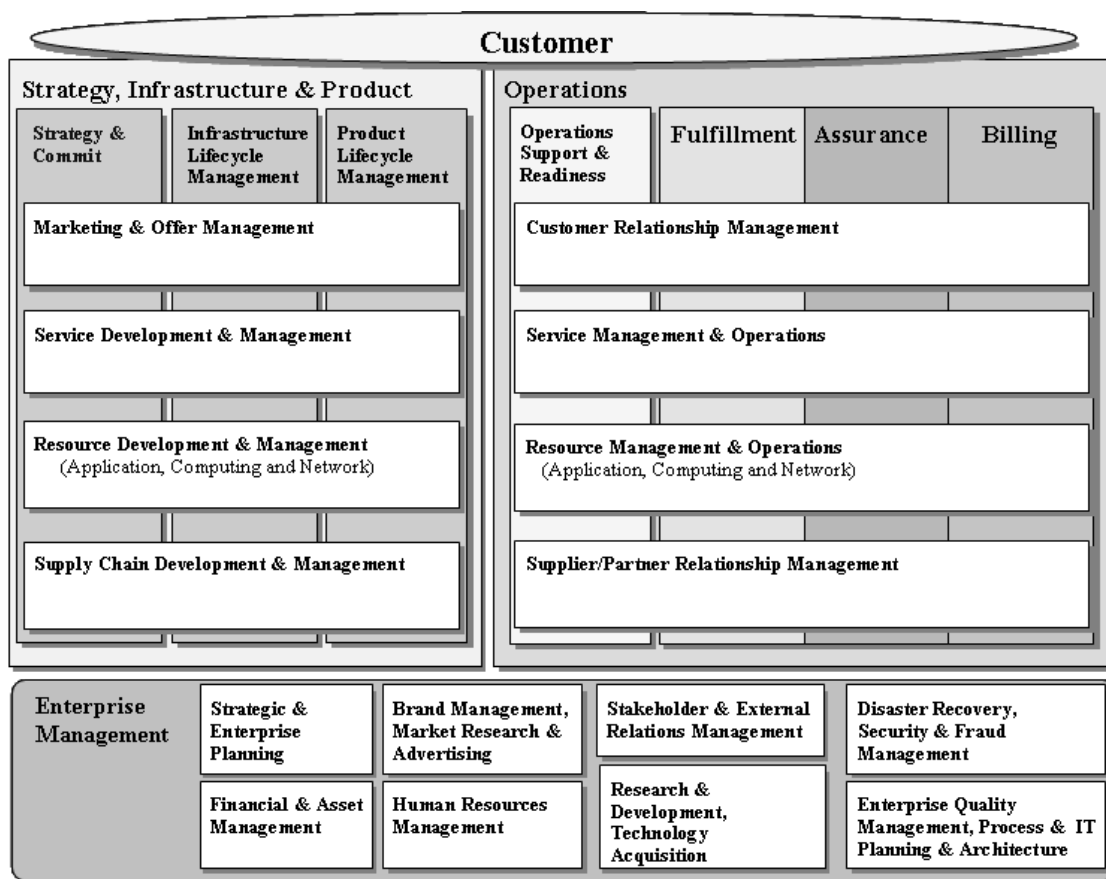


Figura B.4-1: Mapa de Operaciones de Telecomunicaciones Mejorado – eTOM.

Fuente: Enhanced Telecom Operations Map (eTOM) – The business process framework [32]

El área Estrategia-Infraestructura-Producto abarca los procesos involucrados en la elaboración de estrategias a partir de los cuales la empresa planifica el desarrollo y la gestión tanto de la entrega como de la cadena de suministro y el mejoramiento de infraestructura y productos, entendiendo infraestructura no solo como los recursos de tecnologías de la información y de red que soportan el despliegue de productos y servicios, sino también incluyendo la infraestructura operativa y organizacional necesarias para sustentar la comercialización, ventas y procesos de servicios y cadena de suministro.

Por otra parte, el área de Operaciones comprende aquellos procesos que proveen soporte directo al cliente, en ella pueden identificarse cuatro agrupaciones de procesos de extremo a extremo: Aseguramiento, Cumplimiento, Facturación, y

Soporte y Disposición de Operaciones. Ésta última se encarga de asegurar que las tres agrupaciones previas puedan responder a los requerimientos del cliente preparando la información, productos, servicios y recursos, así como los proveedores y socios para entregar y sustentar las instancias de servicio del cliente.

Por último, en el área de Gestión Empresarial se incluyen los procesos de negocio que soportan directamente la administración y el funcionamiento de cualquier empresa, comprendiendo aquellos procesos que están dirigidos hacia el establecimiento y alcance de metas y objetivos corporativos y el soporte de servicios requeridos en la operación de toda la organización. Entre estos procesos se encuentran la Gestión Financiera y la Gestión de Recursos Humanos.

Los procesos pertenecientes a las áreas de Estrategia-Infraestructura-Producto y Operaciones están organizados a su vez dentro de una matriz (nivel 1): Las divisiones verticales representan flujos de proceso extremo a extremo, que atraviesan capas horizontales que representan cada una, interfaces con el cliente, servicios, recursos y socios/proveedores [33]. En cada capa horizontal se ubican agrupamientos de procesos u operaciones más específicos (nivel 2), y dentro de estos bloques se distribuyen las actividades en las cuales se soportan dichos procesos (nivel 3).

Actualmente eTOM es el estándar más ampliamente empleado y aceptado para procesos de negocio en la industria de las telecomunicaciones[34], por lo cual ha sido utilizado dentro del marco de referencia semántico YATOSP, para formular una ontología basada en esta iniciativa de estandarización denominada SeTOM, la cual gracias al carácter de estándar de eTOM y a su amplia aceptación dentro del sector de las telecomunicaciones, se presenta como una alternativa adecuada para abordar el enriquecimiento semántico de los identificadores de las tareas de los procesos de negocio de un operador de telecomunicaciones. La utilización de la ontología *SeTOM* está enmarcada en el enriquecimiento de procesos que están fundamentalmente relacionados con el desarrollo, soporte y despliegue de servicios de telecomunicaciones.

B.4.2 SID (Shared Information/Data Model)

El modelo de Información/Datos compartidos (SID) es un modelo unificado de referencia de datos que proporciona un conjunto único de términos que son comúnmente utilizados en procesos de negocio básicos del dominio de las telecomunicaciones, así como las conexiones y relaciones entre ellos [35]. El modelo SID pretende facilitar la comunicación entre los encargados del negocio, los desarrolladores de software, e incluso los mismos empleados de las empresas, promoviendo la utilización de estos términos para referirse o describir la misma información en los procesos de negocio.

SID captura los conceptos y principios necesarios en la definición de un modelo de información compartida, y define en detalle muchos elementos del negocio (conocidos en SID como “Entidades”) y sus atributos asociados. Una entidad de negocio es un objeto de interés en las empresas tales como cliente, producto, servicios o red, mientras que sus atributos son características que describen la entidad. Juntos, estos términos proporcionan una perspectiva orientada a la información y los datos del negocio que se necesitan para el funcionamiento de la empresa [36].

Para clasificar los datos en una forma usable, la arquitectura de SID está diseñada como un modelo de capas o niveles, las cuales dividen los datos/información en ocho dominios: Mercado/Ventas, Producto, Cliente, Servicio, Recurso, Suministrador/Socio, Empresa y Entidades de Negocio Comunes (Figura B.4-2).

En la capa más superficial, cada uno de los ocho dominios de información está alineado con el marco de referencia eTOM. Dentro de cada dominio existe un alto grado de cohesión entre las entidades del negocio, y entre dominios existe un acoplamiento flexible. Lo anterior facilita la segmentación del problema del negocio en piezas manejables y permite que los recursos se concentren en un área particular de interés. En otras palabras, para un proceso de negocio particular que se esté automatizando, gracias a SID se puede identificar qué información es necesaria dentro de la arquitectura para soportar ese proceso.

Además de contener las definiciones de las entidades empresariales que participan en la gestión de las operaciones de telecomunicaciones, el modelo SID desempeña un papel importante al permitir la integración y la interoperabilidad de los modelos de la iniciativa NGOSS, dado que es el medio por el cual la información útil (es decir la que es relevante para un determinado conjunto de procesos de negocio), puede ser compartida y empleada en múltiples procesos de negocio dentro de todo el sistema [37].

Una de las ventajas que tiene la utilización de SID es que proporciona una fusión de la terminología manejada en las empresas de telecomunicaciones, condensada en un modelo simple de información del negocio, la cual hace posible el desarrollo de aplicaciones OSS/BSS con interfaces verdaderamente interoperables. SID constituye un marco de referencia para la representación de información/datos que pueden ser compartidos y/o reutilizados por aplicaciones OSS/BSS brindados por múltiples proveedores [33].

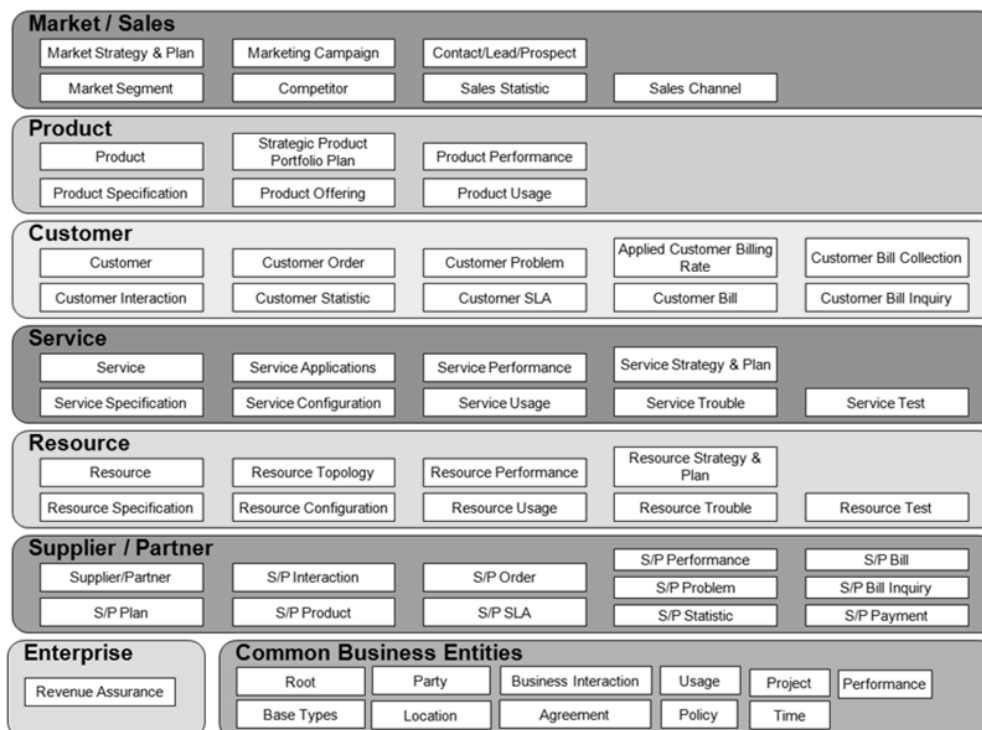


Figura B.4-2. Modelo de Información/Datos Compartidos – SID.
Fuente: Information Framework (SID) In Depth [35]

La ontología SSID propuesta en YATOSP, presenta más de mil conceptos organizados en diez niveles de jerarquía, que conforman un vocabulario común ampliamente utilizado dentro del sector de las telecomunicaciones, los cuales cubren términos como cliente, producto, servicio, recurso, proveedor, entre otros [29]. Por esta razón es recomendable la utilización de esta ontología dentro del presente proyecto, puesto que sus conceptos están estrechamente relacionados con la información que se comparte tanto entre aplicaciones informáticas como entre el personal al interior de un operador de telecomunicaciones. De esta manera es posible realizar el enriquecimiento semántico de las entradas y salidas de las tareas de procesos de negocio con conceptos pertenecientes a esta ontología, dado que estos corresponden a la información que se intercambia para realizar las distintas actividades del negocio.

B.5 Adaptación de Ontologías del Dominio de las Telecomunicaciones

Las ontologías de dominio utilizadas en el presente proyecto fueron definidas dentro del marco de referencia para las telecomunicaciones YATOSP del proyecto SUPER, y están basadas en el conocimiento provisto en los modelos eTOM y SID de la NGOSS. Es importante resaltar que estos modelos han tenido gran aceptación en la industria de las telecomunicaciones, a tal punto que uno de los organismos regulatorios más importantes del sector como es la Unión Internacional de Telecomunicaciones (ITU), los han acogido para la formulación de varias recomendaciones, en el caso de eTOM correspondientes a la serie M.3050 y en el caso de SID está siendo estudiada la propuesta para la recomendación M.3190. Esto refleja que el potencial de estas ontologías es bastante prometedor e involucra a muchos actores de la industria, razón por la cual actualmente las ontologías de eTOM y SID son confidenciales y están siendo estudiadas a fondo para ser modificadas y mejoradas.

De esta manera la adaptación de estas ontologías dentro del presente proyecto no está enfocada en la alteración de los conceptos, instancias y axiomas que las conforman, dado que la construcción y evolución de este tipo de ontologías, son procesos políticos que involucran empresas del sector de telecomunicaciones, entes

regulatorios, grupos de investigación, entre otros. Los cuales unen sus ideales con el fin de estandarizar y masificar el uso de estos recursos en distintas aplicaciones. Por esta razón las ontologías de dominio seleccionadas fueron adaptadas al proyecto específicamente para enriquecer las tareas de los procesos de negocio de telecomunicaciones, permitiendo ampliar el conocimiento respecto a las funcionalidades que desempeñan las tareas, y definiendo puntualmente la ontología más adecuada para enriquecer cada parámetro de las mismas (identificador, entradas y salidas).

En este sentido, de acuerdo a las propuestas de la NGOSS descritas en la sección anterior, se estableció una relación entre los parámetros que describen las tareas de un proceso de negocio y dos de los modelos propuestos dentro de esta iniciativa. En este caso los identificadores de las tareas, los cuales hacen referencia a la operación que esta ejecuta sobre sus entradas para obtener sus salidas, es un parámetro que en el dominio de las telecomunicaciones puede modelarse a partir del conocimiento dispuesto en el estándar eTOM, y la ontología más adecuada para enriquecer semánticamente los identificadores de las tareas es SeTOM, cuyos conceptos e instancias corresponden a verbos o acciones que se llevan a cabo dentro de los procesos de negocio.

Por otro lado, las entradas y salidas de las tareas pueden asimilarse a las entidades que conforman el modelo SID, en tanto corresponden a la información que se comparte entre las tareas de un proceso de negocio, es decir a los objetos del negocio que se intercambian entre dichas tareas. Como consecuencia se definió que la ontología más apropiada para enriquecer semánticamente las entradas y salidas de las tareas es SSID, cuyos conceptos corresponden a sustantivos u objetos propios del dominio de telecomunicaciones.

Anexo C

C Enriquecimiento Semántico de Procesos BPMO

C.1 Introducción

El lenguaje BPOMO (*Business Process Modelling Ontology*) surgió con el propósito de proveer una plataforma estable para la definición de procesos de negocio enriquecida semánticamente, mediante la creación de una ontología capaz de representar artefactos de las múltiples metodologías empleadas para el modelado de procesos de negocio, promoviendo de esta manera, una interpretación única y una representación unificada de los mismos [38].

Las anotaciones semánticas relacionadas con el dominio de un proceso de negocio, son necesarias para llevar a cabo tareas de inferencia sobre los componentes que lo constituyen. La automatización de operaciones como la comparación y el descubrimiento de servicios (servicios descritos a partir de un proceso de negocio), dependen en gran medida de la correcta realización de dichas tareas de inferencia, y por lo tanto, de la pertinencia de las anotaciones semánticas realizadas sobre los elementos que integran el proceso.

La labor de enriquecer semánticamente los componentes de un proceso de negocio se ha abordado a partir de la utilización de la herramienta WSMO Studio, la cual permite modelar gráficamente el proceso en BPMO, al tiempo que genera la

ontología que define el proceso en particular. De esta manera es posible, disponiendo de una ontología de dominio, especificar los conceptos que representan las entradas y salidas de las actividades del proceso.

En este documento da a conocer cómo es posible realizar dichas anotaciones sobre los procesos de negocio, a partir de la implementación de un caso representativo del dominio de las telecomunicaciones. Para esto se dispuso de dos de las ontologías de dominio definidas en el marco de YATOSP: eTOM y SID; y se utilizó el entorno de desarrollo WSMT para facilitar las operaciones de razonamiento sobre la ontología del proceso de negocio de estudio

C.2 Definición del Caso de Estudio y Modelamiento del Proceso.

El caso estudio que se presenta en este documento hace parte de un prototipo implementado dentro del proyecto SUPER correspondiente a un servicio de Gestión Digital de Activos (DAM, *Digital Asset Management*) de *Telefónica*, el cual se encuentra documentado en el entregable 9.5 del proyecto SUPER [39].

De esta manera utilizando la capacidad de modelamiento de procesos de negocio sobre BPMO que provee WSMO Studio, se implementó el subproceso de Autenticación de DAM tal como se muestra en la Figura C.2-1.

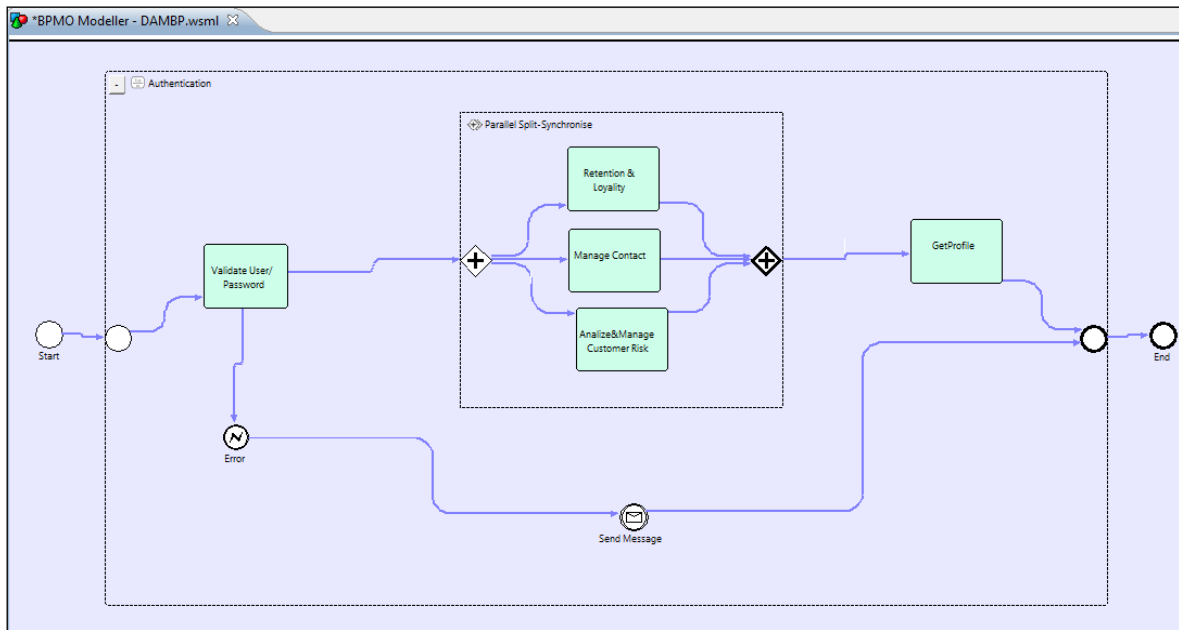


Figura C.2-1: Subproceso de Autenticación – DAM en WSMO Studio.

Una vez modelado el proceso de negocio de esta manera, es posible empezar a enriquecer semánticamente los componentes que lo constituyen. En este caso particular se enriquecen las entradas y salidas de las tareas del proceso.

Para realizar esta labor WSMO Studio cuenta con una *perspectiva* BPMO, la cual pone a disposición las herramientas necesarias para facilitar la edición de los atributos de los elementos del proceso (Figura C.2-2). Para activar esta perspectiva (en caso de no estarlo) es necesario seguir los siguientes pasos:

- ✓ hacer clic en el menú *Window*
 - ✓ Seleccionar *Open Perspective -> Other*
 - ✓ Finalmente seleccionar *BPMO* y hacer clic en *OK*.

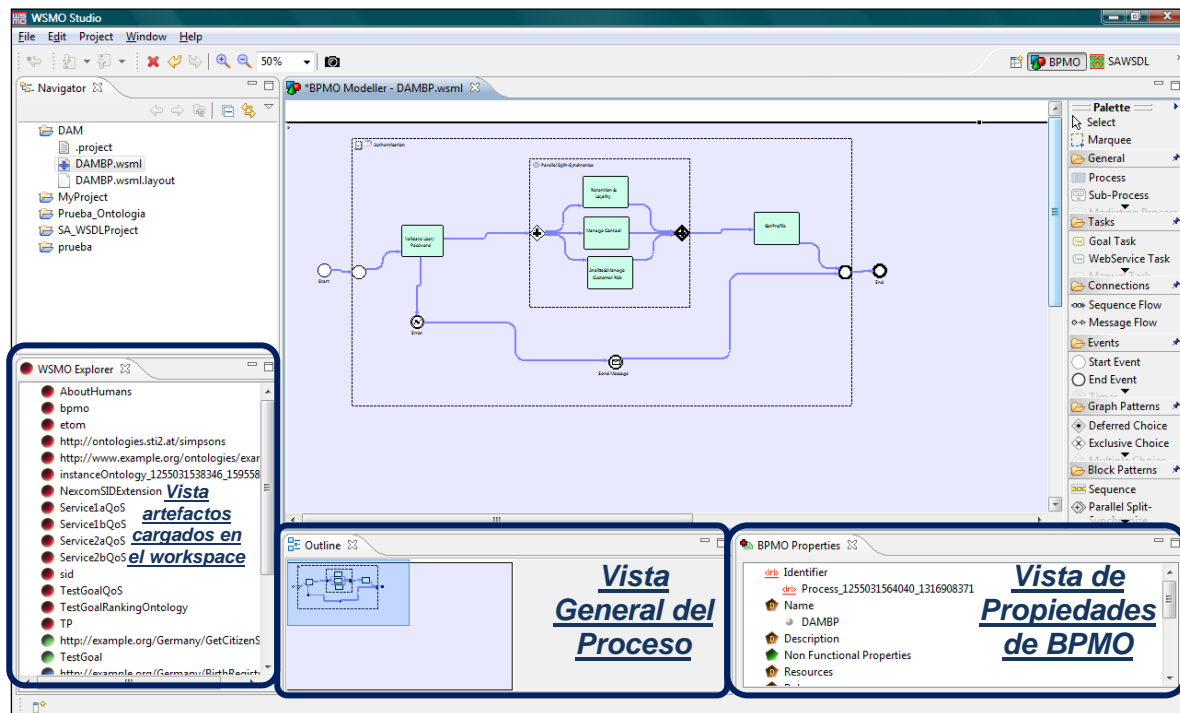


Fig.

Figura C.2-2: Perspectiva BPMO – WSMO Studio.

C.3 Enriquecimiento Semántico de entradas/salidas de las Tareas del Proceso de Negocio

Existen dos formas de enriquecer las entradas/salidas de las Tareas de un Proceso de Negocio: o bien definiendo el concepto que las identifica, o bien definiendo el tipo de dato que las representa. El primer caso se aplica cuando las entradas/salidas representan entidades que hacen parte del conjunto de conceptos una ontología, mientras que el último caso es aplicable cuando las entradas/salidas corresponden a datos básicos, tales como enteros, flotantes, cadenas (string), booleanos, etc.

Para ilustrar lo anterior, considérese la tarea “*Validate User/Password*” del subproceso de Autenticación anteriormente modelado. Esta tarea tiene como entradas dos entidades de tipo String: *CustomerLogin* y *CustomerPassword*. Para establecerlas dentro del modelo, es necesario ubicarse en el diagrama del proceso

en WSMO Studio y seleccionar la tarea. Posteriormente, en la Vista de Propiedades de BPMO (*BPMO Properties*), en la pestaña de *Inputs*, hacer clic derecho y seleccionar *Add Input* (Figura C.3-1). El entorno solicita ingresar el *IRI* que identifica la entrada. En este caso se ingresa el nombre de la entrada como *IRI*, es decir, *CustomerLogin* y *CustomerPassword* (Figura C.3-2).

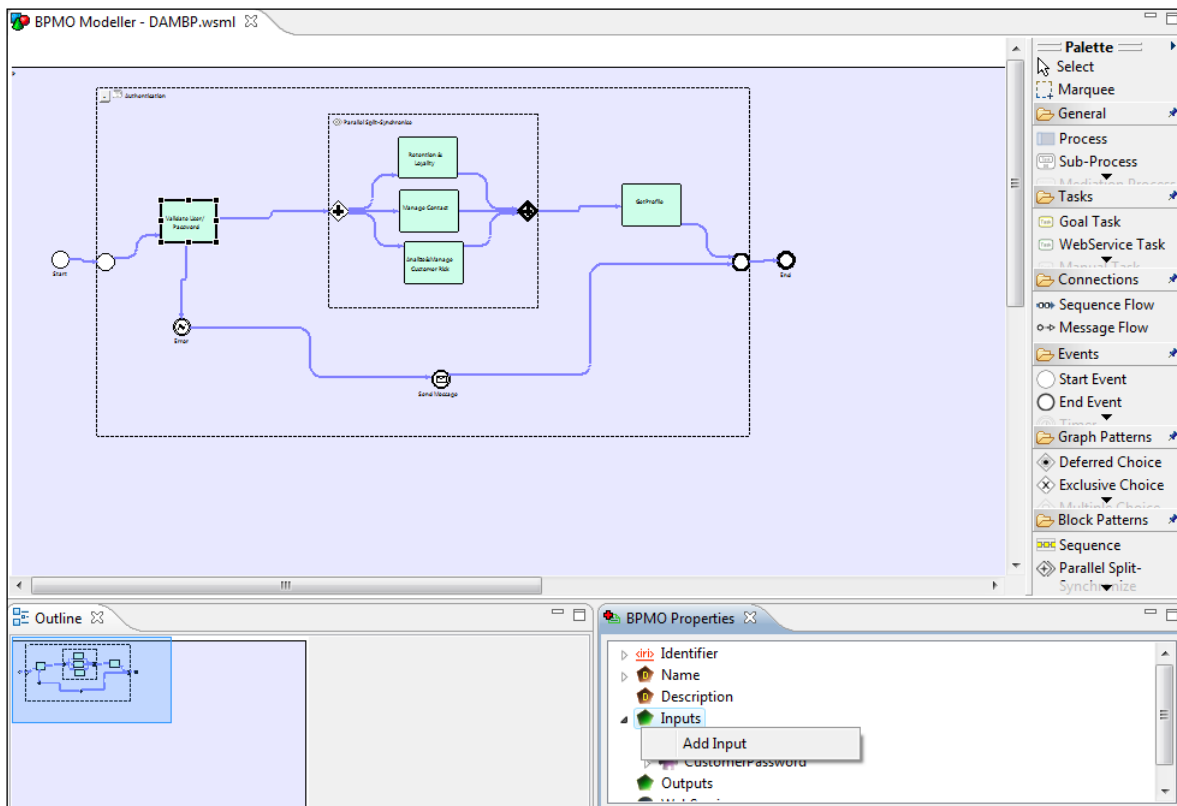


Figura C.3-1: Adición de Entradas a las Tareas BPMO.

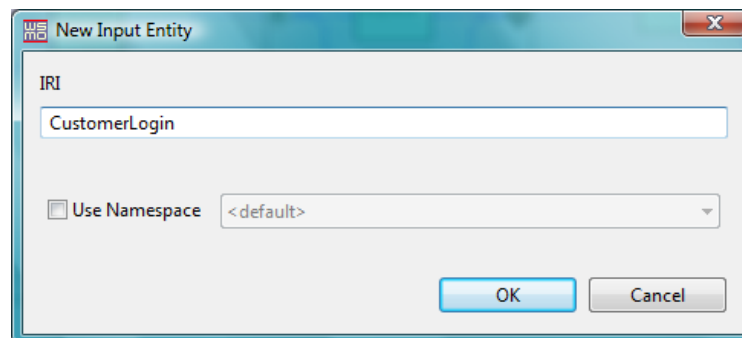


Figura C.3-2: Cuadro de Dialogo Nueva Entrada.

Una vez creadas las dos entradas, es posible visualizarlas en la Vista de Propiedades de BPMO (*BPMO Properties*), tal como se observa en la Figura C.3-3. Además, para especificar el tipo de dato correspondiente a una entrada, es necesario desplegar las propiedades de la misma haciendo clic en la pestaña ubicada a la izquierda del nombre de la tarea. Luego, al hacer clic derecho sobre la propiedad *Data Type*, se selecciona la opción *Set Data Type* (Figura C.3-4).

De esta manera el entorno permite elegir entre varios tipos de datos básicos predefinidos de acuerdo con la especificación de WSMO presentada en el trabajo de Steinmetz y Toma [40]. Para el caso particular que se está considerando, la tarea *CustomerLogin* corresponde con un tipo de dato <http://www.wsmo.org/wsml/wsml-syntax#string> (ver Figura C.3-5).

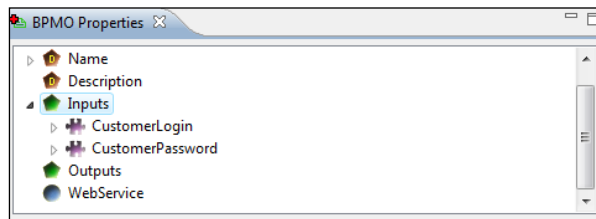


Figura C.3-3: Visualización de entradas de la Tarea.

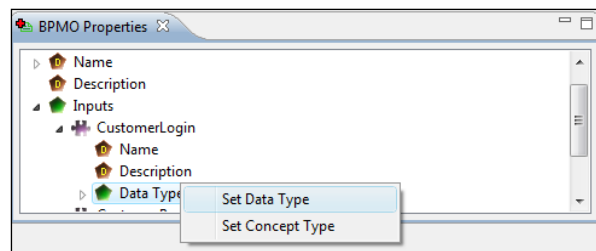


Figura C.3-4: Propiedad Data Type - Entrada CustomerLogin.



Figura C.3-5: Selección de Data Type - Entrada CustomerLogin.

Luego se procede de la misma manera para la segunda entrada definida (CustomerPassword), pero, esta vez se debe tener en cuenta el caso en el cual las entradas/salidas de una tarea representan entidades que hacen parte del conjunto de conceptos de una ontología. Por ejemplo, considérese el caso de la tarea “GetProfile”, la cual tiene como salida una instancia del concepto *PartyProfile* definido en la ontología SID. Para establecer lo anterior dentro del modelo, se procede de la misma forma que se hizo para las entradas, pero esta vez se debe situar en la pestaña *Outputs* de la Vista *BPMO Properties* correspondiente a la tarea *GetProfile*. De esta manera se crea una salida con el nombre *CustomerProfile* tal como se muestra en la Figura C.3-6 Además, es necesario, asegurarse que las ontologías de dominio utilizadas estén cargada dentro de los archivos que hacen parte del proyecto, o en algún otro proyecto activo en el workspace.

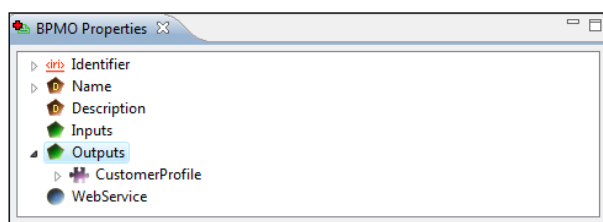


Figura C.3-6: Visualización de salidas de la Tarea.

Para establecer formalmente el concepto al cual pertenece la salida creada, se debe acceder a la propiedad *Data Type*, tal como se realizó para el caso de las entradas de la tarea *Validate User/Password*, solo que esta vez se selecciona la opción *Set Concept Type*, en lugar de *Set Data Type*. Una vez seleccionada esta opción el entorno despliega la lista de ontologías presentes en el proyecto (o en el workspace) permitiendo así escoger el concepto al cual pertenece la salida (o entrada). Para el caso particular que se está abordando, se despliega la ontología SID y se selecciona el concepto *PartyProfile*, tal como se muestra en la Figura C.3-7.

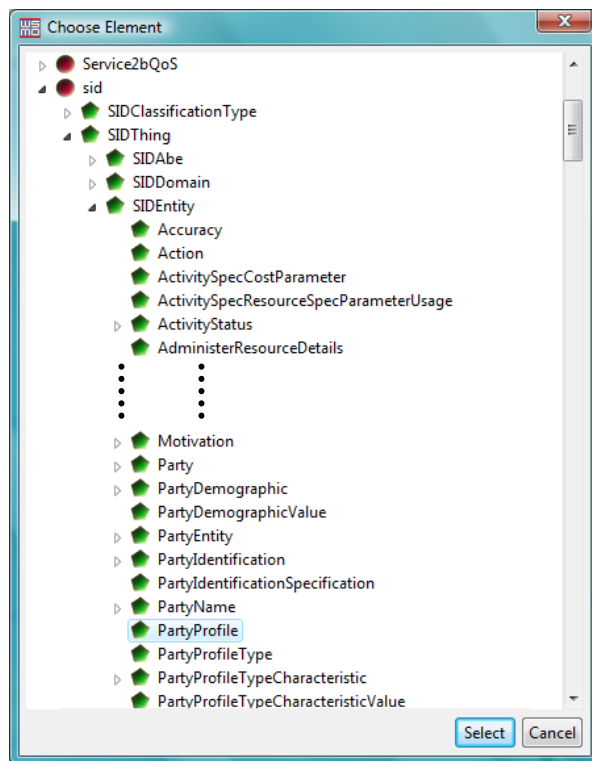


Figura C.3-7: Selección de Concept Type - Salida CustomerProfile.

C.4 Enriquecimiento Semántico de los nombres de las Tareas

Es posible realizar enriquecimiento semántico sobre los nombres de las tareas que componen el proceso de negocio de tal manera que estas se identifiquen con conceptos presentes en una ontología de dominio. Sin embargo, realizar este tipo de anotaciones en WSMO Studio no es un procedimiento tan intuitivo como el que se observó para las entradas/salidas de las tareas, ya que es necesario trabajar directamente sobre el código de la ontología correspondiente al proceso de negocio modelado, de manera que se requiere tener un conocimiento básico de la estructura de una ontología especificada en wsml.

Teniendo en cuenta lo anterior y considerando la orientación hacia el dominio de las telecomunicaciones del presente caso de estudio, se tomó como referencia la ontología eTOM, la cual modela el conocimiento relacionado con los procesos,

actividades y tareas que soportan la operación de una empresa proveedora de servicios de telecomunicaciones. De esta manera, en el caso que se está considerando desde el comienzo de este documento, es posible identificar la Tarea “*GetProfile*” con el concepto *acquireCustomerData* el cual hace parte de la ontología eTOM (es decir, *GetProfile* representa una instancia de este concepto dentro de la ontología del proceso de negocio). Para realizar efectivamente esta anotación, se utiliza el entorno WSMT, el cual además permite realizar algunas operaciones simples de razonamiento. De esta manera es necesario tomar el ontología del proceso (archivo *.wsml) creada en WSMO Studio y pasarla como archivo a un proyecto en WSMT (simple operación de arrastrar y soltar), con el fin de realizar las modificaciones necesarias (Figura C.4-1).

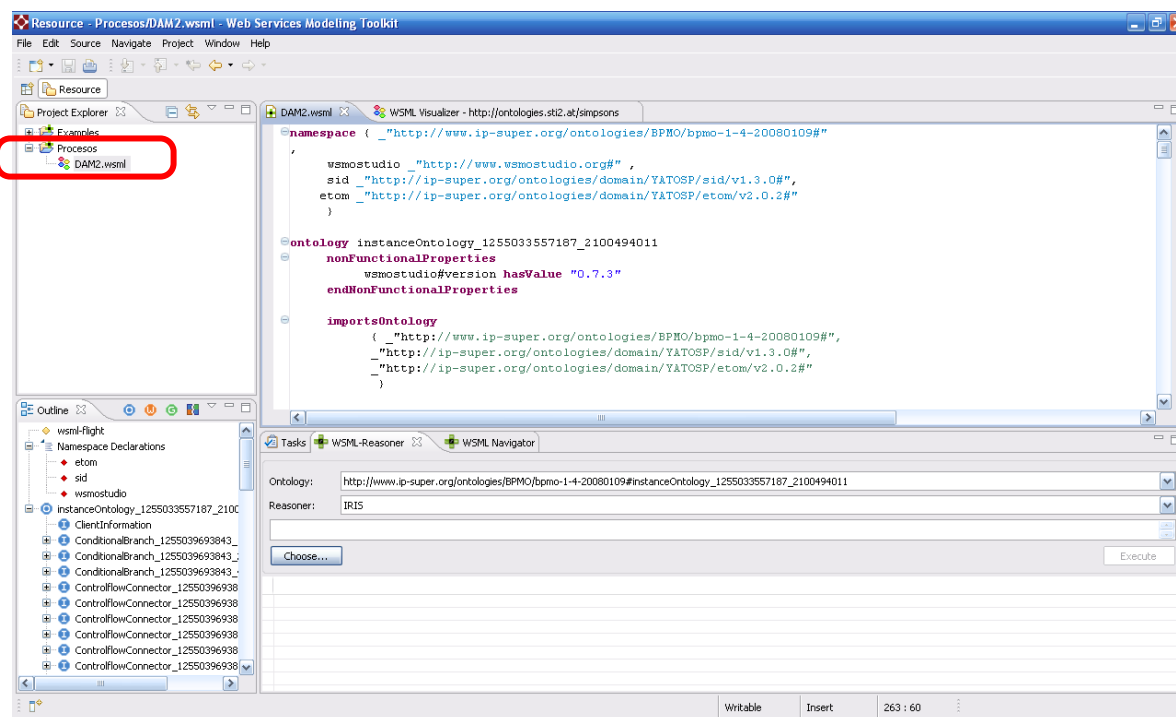


Figura C.4-1: Ontología del Proceso de Negocio de Autenticación en WSMT.

En primer lugar se debe definir la variante de WSMO sobre la cual se va a inferir (en este caso se trata de la variante *flight* dado que el razonador IRIS trabaja sobre archivos WSMO de esta variante), lo cual se puede realizar, como se muestra en la Figura C.4-2, añadiendo la siguiente línea de código al inicio de la ontología del proceso: *wsmVariant _ "http://www.wsmo.org/wsmo/wsmo-syntax/wsmo-flight"*

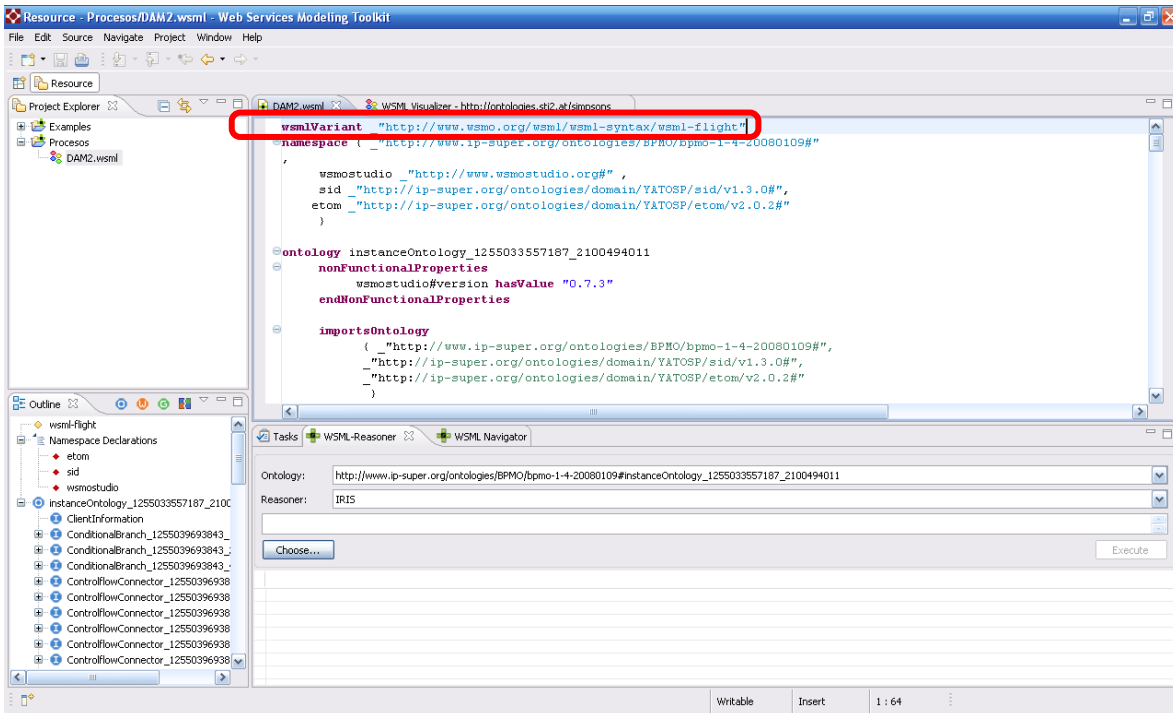


Figura C.4-2: Definición de la variante de WSML.

Posteriormente se debe añadir en el *namespace* y en *importsOntology* las ontologías de dominio que se van a utilizar para realizar las anotaciones, para el caso del presente trabajo estas ontologías son SID y eTOM (Figura C.4-3).

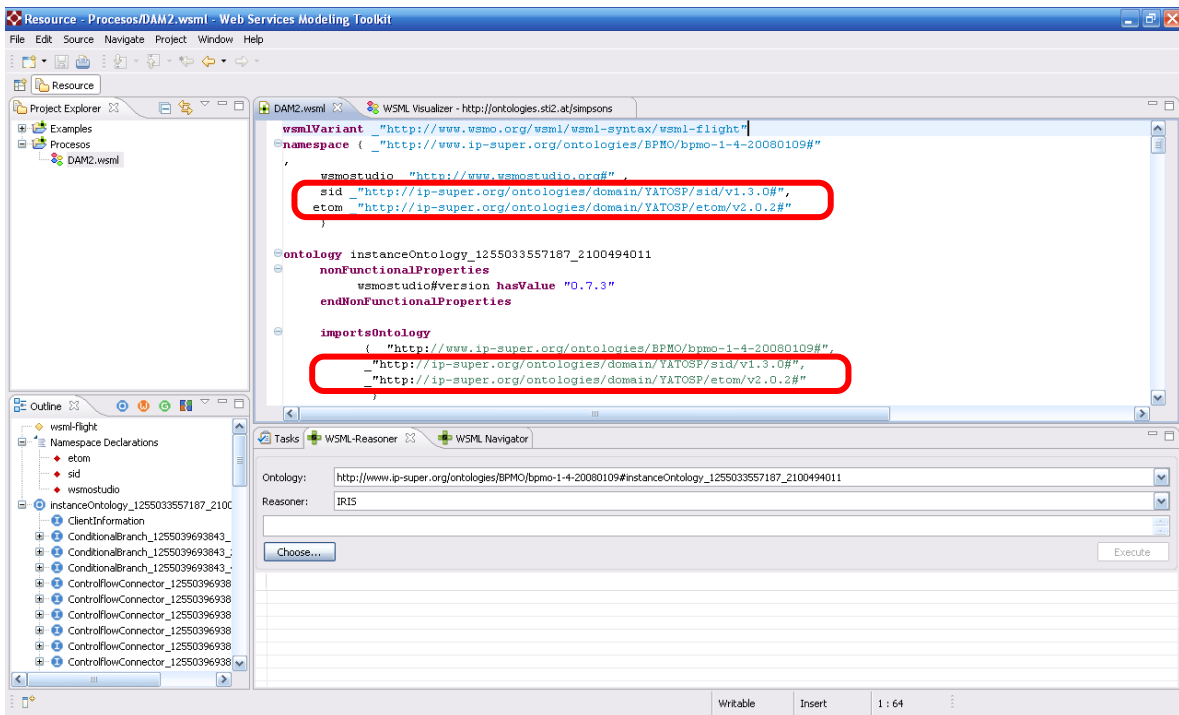


Figura C.4-3: Adición de ontologías de dominio.

Una vez añadidas las ontologías de dominio a la ontología de proceso, se pueden utilizar sus conceptos para realizar las anotaciones semánticas de las actividades. Sin embargo, como se explicó anteriormente en WSMO Studio no es posible realizar dichas anotaciones a los nombres de las tareas, pero en el entorno WSMT se pueden añadir de manera manual como cadenas de texto.

En el caso que se ha venido trabajando en este tutorial se tomó la tarea “*GetProfile*”, la cual puede ser identificada como una instancia del concepto *acquireCustomerData* (que hace parte de la ontología eTOM), por lo tanto se debe añadir a la ontología del proceso con la instancia correspondiente a la tarea “*GetProfile*” el cual es miembro del concepto *acquireCustomerData*. Lo anterior se realiza al modificar la definición de la instancia en *memberOf* adicionando el nombre de la ontología de dominio (como aparece en el namespace) y el concepto de dicha ontología al que se desea concatenar. Para el ejemplo es el concepto *etom#acquireCustomerData* (Figura C.4-4). Es importante anotar que dado que esta instancia ya es miembro del concepto *WebServiceTask*, se debe colocar corchetes al añadir el nuevo concepto de eTOM.

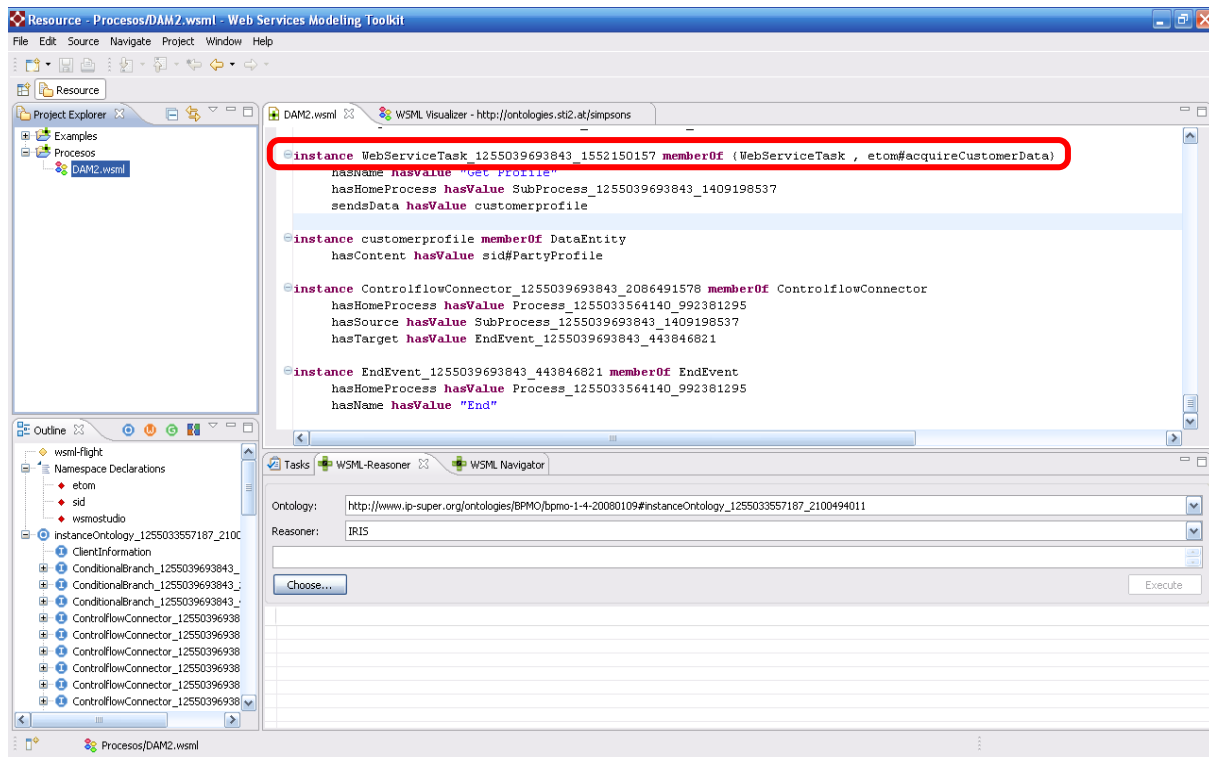


Figura C.4-4: Adición manual de anotaciones semánticas a la tarea Get Profile.

Por último, en la Figura C.4-5 se muestra el resultado de la consulta `?x memberOf ?var0` que se realizó en el razonador IRIS para comprobar la validez del procedimiento anterior. En la misma figura se puede observar que la tarea *Get Profile* cuyo identificador es “*WebServiceTask_1255039693843_1552150157*” es miembro del concepto *acquireCustomerData* de eTOM.

The screenshot shows the WSML Visualizer interface. The main window displays the following instances:

```

instance WebServiceTask_1255039693843_1552150157 memberOf (WebServiceTask , etom#acquireCustomerData)
  hasName hasValue "Get Profile"
  hasHomeProcess hasValue SubProcess_1255039693843_1409198537
  sendsData hasValue customerprofile

instance customerprofile memberOf DataEntity
  hasContent hasValue sid#PartyProfile

instance ControlflowConnector_1255039693843_2086491578 memberOf ControlflowConnector
  hasHomeProcess hasValue Process_1255033564140_992381295
  hasSource hasValue SubProcess_1255039693843_1409198537
  hasTarget hasValue EndEvent_1255039693843_443846821

instance EndEvent_1255039693843_443846821 memberOf EndEvent
  hasHomeProcess hasValue Process_1255033564140_992381295
  hasName hasValue "End"
  
```

The bottom panel shows the query: `memberOf ?var0`. The results table is as follows:

ROW	?x	?var0
20	UnconditionalBranch_1255039693843_84074095	UnconditionalBranch
14	UnconditionalBranch_1255039693843_917614228	UnconditionalBranch
53	WebServiceTask_1255039693843_100514271	WebServiceTask
65	WebServiceTask_1255039693843_1552150157	etom#acquireCustomerData
75	WebServiceTask_1255039693843_1552150157	WebServiceTask

Figura C.4-5: Resultado de la consulta sobre la ontología del proceso.

Anexo D

D Teoría de Grafos

D.1 Introducción

Uno de los aspectos más importantes de la Teoría de Grafos radica en la relevancia que esta materia tiene en la fundamentación matemática de las Ciencias de la Computación, así como en otras disciplinas. Muchos fenómenos discretos pueden ser modelados mediante el uso de la Teoría de Grafos. Además, son de gran importancia para la comprensión de estructuras de datos y en el análisis de Algoritmos. Típicamente, un grafo se representa mediante una serie de puntos (nodos) conectados por líneas (aristas). A continuación se presentan los conceptos más relevantes de la teoría de grafos según [41].

D.2 Definiciones Básicas

Los grafos constituyen una herramienta matemática para modelar sistemas a través de estructuras de datos poderosas y generales que permiten la representación de objetos y conceptos. Los grafos están constituidos por nodos, que representan operaciones o cálculos; y aristas o enlaces entre ellos que representan dependencia de datos [42]. Por esta razón los grafos se pueden considerar como excelentes herramientas para modelar procesos de negocio, en los cuales un nodo puede representar una tarea y las aristas el intercambio de mensajes y datos entre los

nodos, facilitando de esta manera el modelado de estructuras con alto nivel de abstracción, y la representación de relaciones de causalidad entre componentes, flujos de datos, flujos de control y concurrencia en la ejecución de procedimientos paralelos.

En el presente aquellos grafos que representan procesos de negocio se denominan “*grafos de procesos*”. En este tipo de grafos existen nodos simples, que tienen una arista de entrada y otra de salida; y nodos de control, que representan puntos de control de flujo que pueden tener múltiples entradas y salidas dependiendo de su tipo (OR, XOR, AND etc). De esta manera los flujos de ejecución de un grafo de proceso no pueden tener conflictos, pero pueden ser concurrentes. En la Figura D.2-1 se puede observar un grafo en el cual el flujo de datos después del nodo E0 se divide en el nodo de control XOR-Split y se sincroniza en el nodo de control XOR-Join. Entre los cuales dos nodos básicos E1 y E2 toman lugar de manera no determinista.

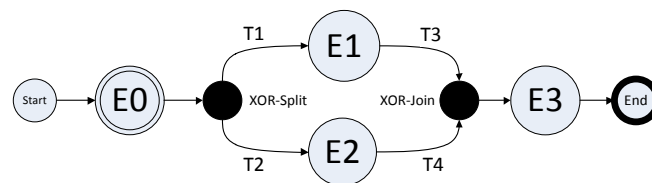


Figura D.2-1: Ejemplo de un grafo de proceso

Formalmente un grafo G es un par (V, E) donde V es un conjunto (llamado conjunto de vértices o nodos) y E un subconjunto de $V \times V$ (conjunto de aristas). Gráficamente los nodos son representados por puntos y las aristas por líneas que los unen. Un nodo puede tener 0 o más aristas, pero toda arista debe unir exactamente 2 nodos. Se denomina orden de un grafo a su número de nodos $|V|$; si $|V|$ es finito se dice que el grafo es finito.

- **Aristas:** si la arista carece de dirección se denota indistintamente $\{a, b\}$ o $\{b, a\}$, siendo a y b los nodos que une. Si $\{a, b\}$ es una arista, a los nodos a y b se les llama sus extremos.

- **Nodos:** dos nodos v, w se dice que son adyacentes si $\{v, w\}$. A (o sea, si existe una arista entre ellos). Se denomina grado de un nodo al número de aristas de las que es extremo. Se dice que un nodo es “par” o “impar” según sea su grado.

- **Caminos:** Sean $x, y \in V$, se dice que hay un camino en G de x a y si existe una sucesión finita no vacía de aristas $\{x, v_1\}, \{v_1, v_2\}, \dots, \{v_n, y\}$. En este caso:

- x e y se llaman los extremos del camino
- El número de aristas del camino se llama longitud del camino.
- Si los nodos no se repiten el camino se dice propio o simple.
- Si hay un camino no simple entre 2 nodos, también habrá un camino simple entre ellos.
- Cuando los dos extremos de un camino son iguales, el camino se llama circuito o camino cerrado.
- Un ciclo es denominado un circuito simple
- Un nodo se dice accesible desde el vértice b si existe un camino entre ellos. Todo vértice es accesible respecto a si mismo

D.3 Clasificación de Grafos

Los grafos se pueden clasificar en dos grupos: *dirigidos* y *no dirigidos*. En un grafo no dirigido el par de nodos que representan una arista no está ordenado. Por lo tanto, los pares (v_1, v_2) y (v_2, v_1) representan la misma arista.

Un grafo No Dirigido $G = (V, E)$ está definido por las siguientes propiedades:

- $V \neq \emptyset$
- $E \subseteq \{x \in \mathcal{P}(V) : |x| = 2\}$

En un grafo dirigido cada arista está representada por un par ordenado de nodos, de forma que representan dos aristas diferentes.

Un grafo Dirigido $G = (V, E)$ está definido por las siguientes propiedades:

- $V \neq \emptyset$
- $E \subseteq \{(a, b) \in V \times V : a \neq b\}$

Dada una arista (a, b) , la letra a representa su nodo inicial y la letra b su nodo final.

A continuación se presentan tres ejemplos de grafos con sus conjuntos de aristas y nodos.

Ejemplo N° 1:

$$G1 = (V1, A1)$$

$$V1 = \{1, 2, 3, 4\} \quad A1 = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$$

Ejemplo N° 2:

$$G2 = (V2, A2)$$

$$V2 = \{1, 2, 3, 4, 5, 6\} \quad A2 = \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 6)\}$$

Ejemplo N° 3:

$$G3 = (V3, A3)$$

$$V3 = \{1, 2, 3\} \quad A3 = \{<1, 2>, <2, 1>, <2, 3>\}$$

Gráficamente estas tres estructuras de nodos y aristas se pueden representar como se muestra en la Figura D.3-1:

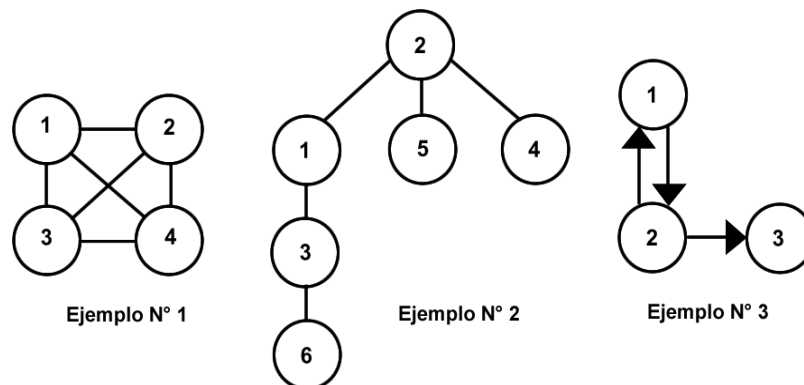


Figura D.3-1: Ejemplos de grafos dirigidos y no dirigidos

Algunos de los principales tipos de grafos se explican a continuación:

D.3.1 Grafo regular

Es aquel grafo con el mismo grado en todos los nodos. Si el grado es k se llamara k -regular. Por ejemplo en la Figura D.3-2, el primero de los siguientes grafos es 3-regular (a), el segundo es 2-regular (b) y el tercero no es regular (c).

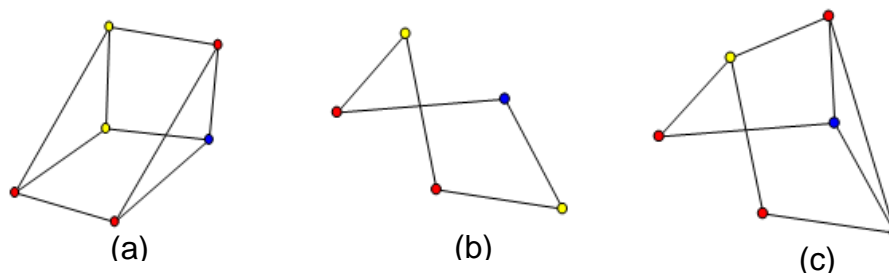


Figura D.3-2: Grafos regulares y Grafo no regular

D.3.2 Grafo bipartito

Es aquel cuyos nodos forman dos conjuntos disjuntos de modo que no haya adyacencias entre nodos pertenecientes al mismo conjunto. En la Figura D.3-3 el primer grafo es bipartito (a) y el segundo no lo es (b).

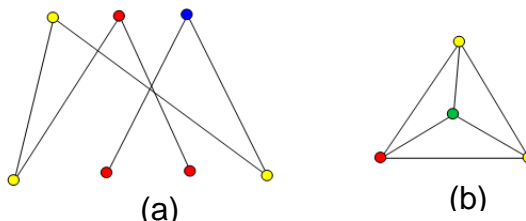


Figura D.3-3: Grafos bipartito y no bipartito

D.3.3 Grafo completo

Es aquel con una arista entre cada par de nodos. Un grafo completo con n vértices se denota K_n . En la Figura D.3-4 (a, b, c, d) se puede observar los grafos K_3 , K_4 , K_5 y K_6 respectivamente.

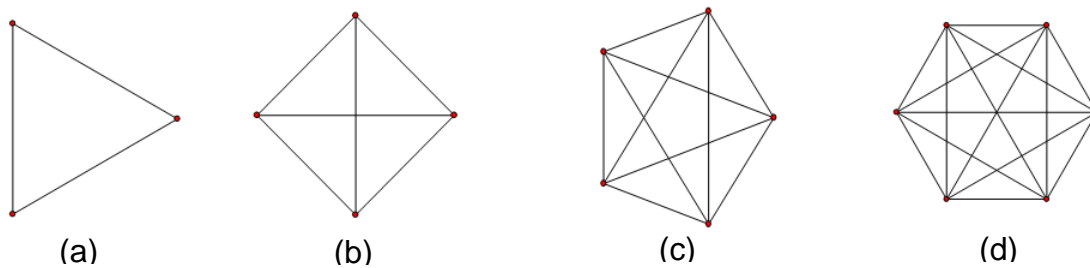


Figura D.3-4: Grafos completos

Todo grafo completo es regular porque cada nodo tiene grado $|V| - 1$ al estar conectado con todos los otros nodos. Además un grafo regular no tiene por qué ser completo. Por otra parte un grafo bipartido regular se denota $K_{m,n}$ donde m, n es el grado de cada conjunto disjunto de nodos.

D.3.4 Grafo nulo

Se dice que un grafo es nulo cuando los nodos que lo componen no están conectados, en otras palabras nodos aislados. En la Figura D.3-5 se muestra un grafo nulo.

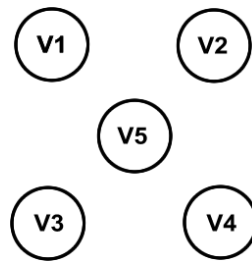


Figura D.3-5: Grafo nulo

D.3.5 Grafos Isomorfos

Dos grafos son isomorfos cuando existe una correspondencia biunívoca (uno a uno), entre sus nodos de tal forma que dos de estos quedan unidos por una arista en común. En la Figura D.3-6 se muestra un ejemplo.

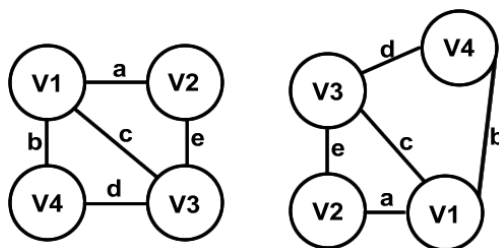


Figura D.3-6: Grafos Isomorfos

D.3.6 Grafos Conexos:

Un grafo se puede definir como conexo si cualquier nodo V pertenece al conjunto de vértices y es alcanzable por algún otro. En otras palabras un grafo conexo es un grafo no dirigido de modo que para cualquier par de nodos existe al menos un camino que los une. En la Figura D.3-7 se muestra un ejemplo.

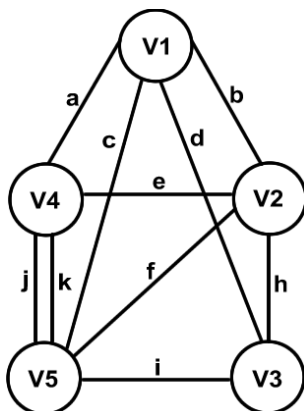


Figura D.3-7: Grafo Conexos

D.3.7 Árboles

Un árbol se define como un tipo de grafo que no contiene ciclos, es decir un grafo acíclico y conexo. En la Figura D.3-8, se observan ejemplos de árboles, como se puede notar ninguno de ellos contiene repeticiones (ciclos).

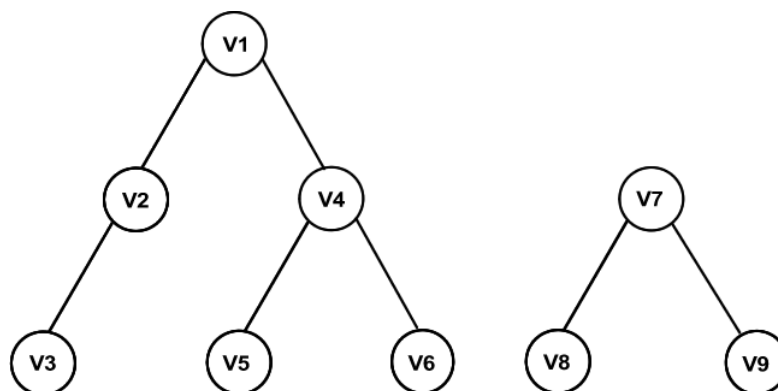


Figura D.3-9: Árboles

D.4 Recorrido de un Grafo

Se denomina recorrido a un grafo cuando se alcanzan todos los nodos que estén relacionados con uno denominado nodo de salida. Existen básicamente dos técnicas para recorrer un grafo: el recorrido en anchura; y el recorrido en profundidad.

D.4.1 Recorrido en anchura

El recorrido en anchura supone recorrer el grafo, a partir de un nodo dado, en niveles, es decir, primero los que están a una distancia de una arista del nodo de salida, después los que están a dos aristas de distancia, y así sucesivamente hasta alcanzar todos los nodos a los que se pudiese llegar desde el nodo salida. La búsqueda en anchura (en inglés BFS - Breadth First Search) es un algoritmo para recorrer o buscar elementos en un grafo (usado frecuentemente sobre árboles). Intuitivamente, se comienza en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo) y se exploran todos los vecinos de este nodo. A continuación para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el árbol. Formalmente, BFS es un algoritmo de búsqueda sin información, que expande y examina todos los nodos de un árbol

sistemáticamente para buscar una solución. El algoritmo no usa ninguna estrategia heurística.

D.4.2 Recorrido en profundidad

El recorrido en profundidad busca los caminos que parten desde el nodo de salida hasta un punto en el cual no es posible avanzar más. Cuando ya no puede avanzarse más sobre el camino elegido, se vuelve atrás en busca de caminos alternativos, que no se estudiaron previamente. Un recorrido en profundidad (en inglés DFS - Depth First Search) es un algoritmo que permite recorrer todos los nodos de un grafo o árbol de manera ordenada, pero no uniforme. Su manera de funcionar se basa en ir expandiendo cada una de los nodos que va localizando, de manera recursiva, recorriendo todos los nodos de un camino concreto. Cuando ya no quedan más nodos por visitar en este camino, regresa hacia atrás (backtracking), de tal manera que comienza el mismo proceso con cada uno de los hermanos del nodo ya procesado. El recorrido en profundidad acepta tres tipos de recorridos al grafo o el árbol, el recorrido pre-orden, recorrido post-orden y el recorrido in-orden.

El recorrido pre-orden se hace desde el primer nodo hasta los hijos siguiendo el orden: nodo raíz, nodo izquierda, nodo derecha. El recorrido post-orden visita primero los hijos siguiendo este orden: Nodo izquierda, nodo derecha, nodo raíz. El recorrido in-orden primero visita algún hijo, luego al nodo padre y por ultimo al otro hijo.

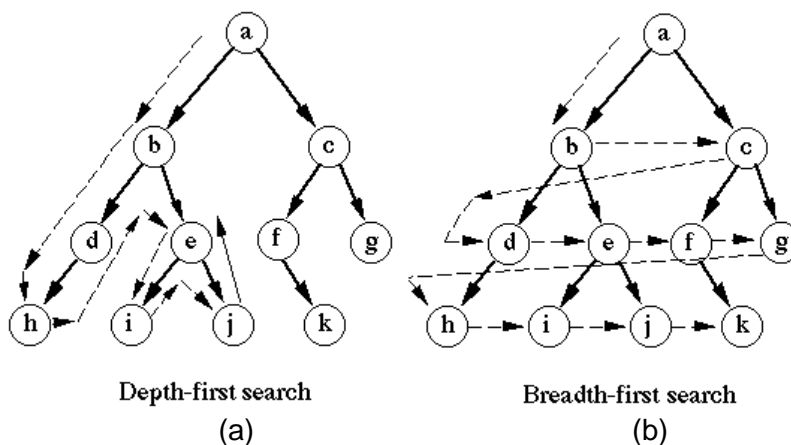


Figura D.4-1: Recorrido en profundidad y en anchura
Fuente: [43]

Por ejemplo, para el árbol de la Figura D.4-1(a) los recorridos en profundidad (Depth-first search) son los siguientes:

- *Recorrido pre orden:* b, d, h, e, i, j, c, f, k, g.
- *Recorrido post orden:* h, d, i, j, e, b, k, f, g, c, a.
- *Recorrido in orden:* h, d, b, i, e, j, a, f, k, c, g.

Mientras que el recorrido en anchura (Breadth-first search) a grafos o árboles se hace nivel a nivel, como se muestra en la Figura D.4-1(b) y su recorrido es el siguiente: a, b, c, d, e, f, g, h, i, j, k.

D.5 Isomorfismo de Grafos

El trabajo de Okkyung y Sangyong [44] define el isomorfismo de grafos de la siguiente manera: Considérese dos grafos G y G' ; un isomorfismo de grafos entre G and G' es un mapeo biyectivo $f: V \rightarrow V'$ tal que $\alpha(v) = \alpha'(f(v)) \forall v \in V$. Para cualquier arista $e = (u, v) \in E$ existe una arista $e' = (f(u), f(v)) \in E'$ tal que $\beta(e) = \beta'(e')$ y para cualquier arista $e' = (u', v') \in E'$ existe una arista $e = (f^{-1}(u'), f^{-1}(v')) \in E$ tal que $\beta(e) = \beta'(e')$. Si $f: V \rightarrow V'$ es un isomorfismo entre los grafos G y G' , y G' es un subgrafo de otro grafo G'' , es decir $G \subset G'$, entonces la función f es llamada un isomorfismo de subgrafos de $G \subset G'$.

El trabajo de Zampelli et al [45] define el isomorfismo de sub-grafos así: Dados dos grafos $Q = (V(Q), E(Q), L_v, F)$ y $G = (V(G), E(G), L_v', F')$, se dice que Q es un sub-grafo isomorfo de G , si y solo si existe una función $g: V(Q) \rightarrow V(G)$, tal que $\forall v \in V(Q), F(v) = F'(g(v))$ y $\forall v_1, v_2 \in V(Q), \overrightarrow{v_1, v_2} \in E(Q) \Rightarrow \overrightarrow{g(v_1), g(v_2)} \in E(G)$.

En otras palabras dos grafos son isomorfos si tienen el mismo número de nodos y la misma estructura. En la Figura D.5-1(a) se presenta un ejemplo de un isomorfismo de grafos en el cual la función biyectiva F está dada por la función $F = \{ (1, A), (2, E), (5, D), (4, B), (3, C) \}$. Mientras que en la Figura D.5-1(b) se

presenta un ejemplo de un de sub-grafo isomorfo donde F puede tomar dos caminos: $F = \{(A1, A1), (B2, B2), (C3, c3)\}$ y $F = \{(A1, A4), (B2, B2), (C3, c3)\}$

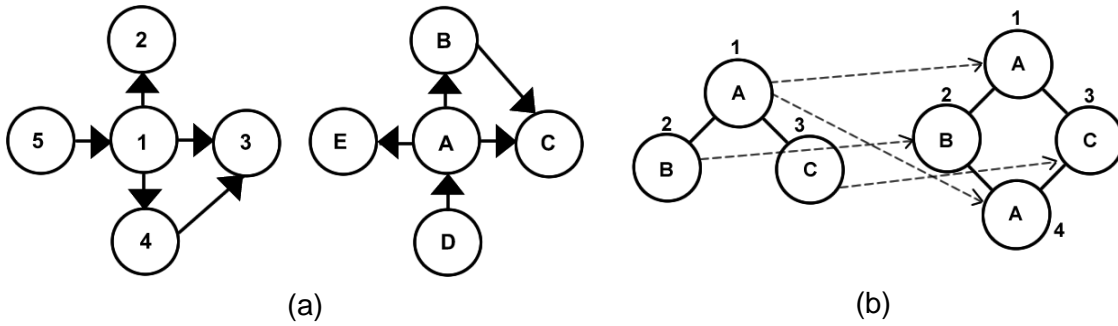


Figura D.5-1: Ejemplo grafo isomorfo (a) y sub-grafo isomorfo (b)

Anexo E

E Repositorios de Procesos de Negocio

E.1 Introducción

En este anexo se presentan las propuestas de repositorios de procesos disponibles actualmente. Un repositorio digital según [46] se define como un sitio centralizado donde se almacenan y se mantiene recursos digitales (textuales, imágenes, sonidos). En este contexto el término repositorio hace referencia a un lugar donde se almacenan servicios Web o procesos de negocio.

E.1.1 Repositorios de Servicios Web

Actualmente existen varios repositorios de servicios Web disponibles en línea presentados como una organización de servicios publicados en páginas Web. Estos tienen motores de búsqueda que realizan las búsquedas de acuerdo al nombre del servicio. Mediante la búsqueda de una palabra clave que retorna el servicio que contiene esa palabra como una sub cadena incluida en los nombres y descripciones de estos servicios Web. La búsqueda también tiene en cuenta los parámetros de entrada y salida de los servicios y los mensajes que se intercambian. Entre los más importantes están: UDDI , Bindingpoint [47], .NET XML Web Services Repertory [48], Woogle [49], eSynaps [50].

Existen otros repositorios que ofrecen categorías de búsqueda que indican el dominio de los servicios, por ejemplo (Negocios, Comercio, Comunicaciones, Salud, etc.), es decir, permiten realizar las búsquedas de los servicios en forma alfabética así como también los nombres y las descripciones de los servicios. Entre estos se encuentran: WebService Resource[51], Web Service List [52], WebserviceX.NET [53] y [54]

E.1.2 Repositorios de Procesos de Negocio

A continuación se nombran y se describen algunos de los repositorios de procesos disponibles en el mercado. Algunos autores como Yan et al [55] presentan los resultados del análisis detallado de 16 repositorios de procesos de negocio, en donde se comparan de acuerdo a sus características de datos, funcionalidad y gestión; y además tienen en cuenta los aspectos de almacenamiento, recuperación, integración e indexación. En esta sección se describen dichos repositorios.

- **Business Process Library (BPL)** [56] es el resultado del primer diseño y prototipo de la librería de procesos de negocio en el proyecto SUPER. La BPL provee las operaciones estándar de un Sistema de Gestión de Base de Datos tales como el soporte de transacciones para la manipulación de datos. Asimismo, permite crear, leer, actualizar, borrar (CRUD) artefactos utilizando una herramienta de modelado definida en el proyecto SUPER. Realiza dos tipos de consultas, la primera en la cual un sistema de base de datos tradicional puede procesar [57] y la segunda permite consultas semánticas haciendo inferencia sobre la información contenida en una base de datos relacional a través de una máquina de razonamiento ontológico [58]. Además, almacena modelos de procesos descritos usando las siguientes ontologías: BPMO, sBPEL [57]. BPL es un proyecto muy amplio que comprende una plataforma para la gestión de procesos de negocio, el inconveniente es que está en fase de desarrollo y en algunos casos no hay acceso para algunas de las funcionalidades del mismo, además tiene una arquitectura cerrada y utiliza herramientas propietarias del proyecto SUPER.

- **ebXML (Electronic Business using eXtensible Markup Language) Registry/ Repository** [59] es un repositorio incluido en la familia de estándares ebXML, el cual empezó como una iniciativa de OASIS (Organization for the Advancement of Structured Information Standards) y UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business). Este repositorio permite almacenar cualquier tipo de dato incluyendo: descripciones de servicios Web, documentos y datos XML, datos binarios (imágenes, archivos de sonido, datos de video, archivos de aplicaciones ejecutables, entre otros). Mediante el registro se puede adicionar meta información a los datos ingresados en el repositorio, además se puede hacer búsquedas y clasificaciones mediante un mecanismo avanzado de consultas que pueden ser filtradas en consultas XML o SQL [60]. Puede interactuar con otros lenguajes de procesos como lo son WS-Choreography Description Language (WS-CDL) para la coreografía de servicios Web y con WS-Business Process Execution Language (WS-BPEL) para la orquestación de servicios Web [61]. Este repositorio no tiene capacidades de consulta semánticas, solamente SQL y utiliza un lenguaje de definición de procesos (ebBP) propio de los estándares ebXML.
- **IBM BPEL Repository** [62] es una parte del proyecto Business Process Integration and Automation (BPIA) del laboratorio de investigaciones de IBM en Zurich [63]. El repositorio almacena procesos de negocio BPEL y otros documentos XML en sistemas de archivos. Permite hacer consultas a los archivos XML como objetos EMF (Eclipse Modeling Framework) utilizando un lenguaje de consultas orientado a objetos (OCL) así como también proporciona operaciones básicas como crear, leer, escribir y borrar para manipular los datos almacenados en XML como objetos Java. Aunque el usuario manipula los datos como objetos, la información es almacenada en forma transparente en el correspondiente documento XML descrito por lenguajes como: BPEL, WSDL, XSD, entre otros [64]. Es importante resaltar que permite manipular diferentes documentos XML y que la arquitectura del repositorio está creada para ser adaptada y añadirle nuevas funcionalidades tales como un nuevo motor de consultas OCL y otras extensiones EMF. Solamente permite almacenar documentos descritos en BPEL y XML y no tiene características de búsqueda semánticas y las consultas se realizan a través de objetos OCL, necesitando un motor de OCL, muy ligado al IDE Eclipse.

- **IPM (Integrated Process Management)** [65], (Repositorio para la Gestión Integrada de Pprocesos), gestiona procesos a través del ciclo de vida. Como tal, no sólo tiene funciones básicas de almacenamiento y recuperación de modelos de procesos de negocio, también cuenta con funciones avanzadas en la gestión de configuración y de versiones. La información de los procesos se puede intercambiar con el repositorio a través del lenguaje IPM Executable Process Definition Language (IPM-EPDL)² [66]. Internamente, el repositorio tiene conectores para almacenar información de los procesos en una base de datos XML o una base de datos relacional. IPM se centra en el almacenamiento de los modelos de procesos específicos de una compañía procedimientos y la ejecución de las instancias de esos procesos. El repositorio IPM tiene un buen soporte para almacenar y recuperar los modelos de procesos e información relacionada. Para ello un lenguaje de consulta, el IPM Proceso Query Language (IPM-PQL), el cual tiene soporte para las consultas específicas en los procesos como la búsqueda de procesos que tienen una determinada actividad o cierta transición de una actividad a otra. IPM también soporta la búsqueda de procesos en el repositorio utilizando una serie de clasificaciones. Tanto el lenguaje de modelado como de consulta es propietario y está diseñado para modelar los procesos específicos de las compañías.
- **RepoX** [67] es un repositorio de modelo de proceso basado en XML, el cual es parte del sistema de workflow llamado METEOR³. Aunque forma parte del proyecto METEOR, se ha desarrollado específicamente con la intención de estandarizar el intercambio de modelos de procesos entre una herramienta de definición de proceso y un motor de workflows. Los modelos de procesos pueden ser intercambiados usando documentos XML en un formato predefinido. Internamente los modelos se almacenan en una base de datos orientada a objetos. RepoX almacena los aspectos del flujo de control de los procesos junto con los datos que son usados en los procesos y los roles que son autorizados para realizar tareas en los procesos. Desde que RepoX fue definido como un

² IPM-EPDL son las siglas de IPM Executable Process Definition Language es un formato basado en XML que se puede utilizar para almacenar información sobre los modelos de procesos y actividades, el flujo de control, la estructura organizacional, la autorización y asignación de recursos, datos y monitoreo.

³ El proyecto (METEOR) fue desarrollado para la gestión de procesos de workflows para gran escala, para aplicaciones de workflows en entornos de computación heterogéneos y multi-empresariales.

repositorio para motores de workflows, éste almacena modelos de procesos específicos para compañías. No cuenta con una notación específica, la descripción de los procesos se realiza a través de DTD, la cual tiene bastantes limitaciones en la definición de elementos de contenido, por tal motivo surgieron otros lenguajes de esquemas como lo son XMLSchema. Además las consultas están basadas en XQuery el cual permite extraer información en documentos XML, pero no permite realizar consultas semánticas.

- **BPMN Repository** [68] es la arquitectura de un repositorio, más que una implementación completa. Está dirigido específicamente a repositorios de modelos de procesos que soporten procesos inter organizacionales, esto es, procesos o composiciones de procesos múltiples que abarquen términos organizacionales. Esta arquitectura propone un amplio rango de tipos de procesos y de aspectos que deberían ser soportados para ese propósito, tales como los aspectos de flujo de control, aspectos de datos, la estructura organizacional y la información de autorización, monitoreo y controles. En cuanto a los tipos de procesos que debería soportar, están: modelos de referencia, modelos específicos de compañías, instancias de procesos e información histórica de las instancias de los procesos. Además plantea que los procesos y la información relacionada, deberían ser almacenadas en un formato XML estándar. Solamente describe una arquitectura que combina varios trabajos y presenta un repositorio distribuido en capas, el lenguaje de modelado es BPMN, no tiene un lenguaje de consulta definido y no permite consultas semánticas.
- **ORYX** [69] es una herramienta de modelado de procesos basada en Web que soporta búsqueda de usuarios, creación, almacenamiento y actualización de modelos de procesos en línea. La herramienta utiliza un repositorio para almacenar los modelos de procesos de negocio que son creados en él. ORYX se enfoca principalmente en el aspecto del flujo de control y almacena modelos de procesos específicos de compañías, además soporta muchas notaciones de modelado de procesos, incluyendo: BPMN, EPC, redes de Petri, redes de workflows, FMC Block Diagrams⁴ y XForms⁵. Internamente, los procesos son

⁴ Los diagramas de bloques FMC son utilizados para realizar bosquejos de composiciones estructurales de sistemas a través de una notación gráfica que pueda ser fácil de entender.

⁵ XForms es un formato XML diseñado por el W3C para definir interfaces de usuario, principalmente formularios Web.

almacenados en una base de datos, externamente representa modelos de procesos en un formato RDF. Además no solamente se enfoca en las funciones básicas de almacenamiento (CRUD), sino que también implementa funciones de importación y exportación de modelos en formatos ERDF⁶, JSON⁷, PNML⁸, XPDL y el formato XHTML se puede transformar en redes de Petri. Para la consulta de procesos, ORYX integra BPMN-Q. El almacenamiento y recuperación de procesos se hace a través de RDF y no soporta consultas semánticas, se hace necesario combinar RDFSchema y OWL para añadir semántica al contenido.

- **BP-Suite** [70] es un conjunto de herramientas basadas en la ejecución de procesos especificadas en BPEL. Consta de BP-QL (un lenguaje para las definiciones de procesos), BP-Mon (una herramienta para monitorear las instancias de procesos de negocio en ejecución) y BP-Ex (una herramienta para analizar las anotaciones de las ejecuciones). BP-Suite soporta los aspectos de actividades, flujo de control, datos, recursos, estructura organizacional, monitoreo y autorización, además del almacenamiento de modelos de procesos, instancias de procesos e información histórica acerca de las instancias. Los modelos de procesos son almacenados en formato BPEL. Además de las funcionalidades estándar de una base de datos para el almacenamiento y recuperación de modelos de procesos, BP-Suite soporta consultas sobre los 3 tipos de procesos que puede almacenar (definiciones, instancias en ejecución, anotaciones). Solamente utiliza BPEL como lenguaje de modelado de procesos, BP-QL como lenguaje propietario para consultas, instancias y manejo de historial a través del IDE Eclipse, necesita del IDE Eclipse para ser ejecutado y tampoco permite realizar consultas semánticas.
- **Process Handbook** [71] (El manual de procesos) es una base de conocimiento de descripciones de procesos; este consta de: Un esquema para organizar modelos de procesos y sus actividades, una colección de modelos de procesos que está organizado de acuerdo a ésta clasificación y una colección de herramientas para la gestión de la base de conocimiento. El enfoque principal del Process Handbook es organizar el conocimiento relacionado a los procesos y no

⁶ ERDF es una sintaxis para escribir HRML de una manera que la información en un documento HTML extraída en RDF.

⁷ JSON es el acrónimo de JavaScript Object Notation, el cual es un formato ligero para el intercambio de datos.

⁸ PNML acrónimo de Petri Net Markup Language, es un formato de intercambio estándar basado en XML para redes de Petri.

proveer información relacionada a los modelos de procesos. Consecuentemente, el repositorio se basa en texto más que en modelos y el meta modelo principalmente contiene clases que almacenan modelos de procesos y sus descripciones para relacionar los modelos de procesos uno a uno y con sus respectivos elementos. Los modelos de procesos se almacenan a través de herramientas propietarias, en las cuales la información se ingresa en forma de texto. Internamente los modelos de procesos son almacenados en una base de datos genérica. Solamente se enfoca en almacenar las referencias a los procesos y tenerlos disponibles como una base de conocimiento generado, además que solamente se hace una descripción de los procesos en lenguaje natural (texto) y no especifica un tipo de almacenamiento determinado.

- **Process Reuse Architecture** [72] (La arquitectura de reutilización de Procesos), es una arquitectura y una herramienta para almacenar modelos de procesos reutilizables, ya sea en parte o en su totalidad. El principal tipo de proceso de la arquitectura consiste en procesos de referencia configurables llamados procesos del entorno. Para configurar un proceso de referencia, la arquitectura también almacena patrones de procesos y modelos de procesos específicos de una compañía. Estos pueden ser reutilizados en parte o totalmente, para adaptar un proceso de referencia configurable a los requerimientos de una organización en particular. Así como *The Process Handbook*, esta arquitectura se enfoca en describir los procesos más que en el modelado de los mismos. Sin embargo, esta arquitectura tiene un conjunto más elaborado de conceptos para la creación de estas descripciones, cubriendo aspectos como: características del proceso, organización, estructura y relación de los modelos de procesos a controles y objetivos estratégicos. Los diferentes tipos de procesos pueden ser almacenados en formato XML, la búsqueda de procesos se realiza basada en texto. Al igual que el anterior solamente hace descripciones de los procesos en lenguaje natural (texto) y no se enfoca en el modelado.
- **Library for Process Programing** [73] (La librería para la programación de procesos), es una arquitectura para organizar modelos de procesos de negocio en librerías, en las cuales los modelos de procesos de negocio individuales pueden ser reutilizados. Esta librería se enfoca en la definición de procesos, representación de instancias de procesos y adaptación de los procesos para su

reutilización. Con el propósito de definir procesos, el sistema introduce un lenguaje llamado “P-language”, que se parece a un lenguaje de programación textual. Este se enfoca en la definición de procesos y actividades, flujo de control y datos utilizados por los procesos. La notación de los modelos así como la recuperación de los mismos, utiliza un lenguaje propietario denominado “P-language”, para el almacenamiento utiliza sistemas de archivos.

- **Repository for workflows systems** (El repositorio para sistemas de workflows) [74], es un proyecto que provee un modelo conceptual para almacenar modelos de workflows, los requerimientos para gestionar repositorios de modelos de workflows y una arquitectura diseñada para implementar un gestor de repositorio. Este repositorio soporta los aspectos de actividades, flujo de control, datos, recursos y monitoreo, además almacena workflows en una base de datos orientada a objetos. Esta base de datos provee las funcionalidades básicas para almacenar y recuperar workflows. No especifica un lenguaje de modelado ni tampoco un lenguaje para el almacenamiento y recuperación de los workflows.
- **ProcessGene** [75] es un proyecto que provee una herramienta para la consulta de modelos de procesos de negocio. Consta de 4 partes: Ámbito asistencial (SA), Interface de especificación de consultas (QSI), un intérprete de consultas (QI) y un empaquetador de resultados de consultas (QRP). Los usuarios de ProcessGene proveen los alcances de las preguntas y las especificaciones a través de los módulos SA y QSI, entonces el módulo QI compila la especificación a los requerimientos de consulta y el módulo QRP retorna los resultados de la consulta. ProcessGene soporta las funcionalidades estándar de almacenar y recuperar modelos de procesos. No utiliza un lenguaje específico de modelado y las consultas se realizan a través de SQL, por lo tanto no permite realizar consultas semánticas.
- **Process Variant Repository (PVR)** [76] provee mecanismos para tratar con diferentes variantes de un modelo de procesos de negocio en tiempo de ejecución. Para este propósito contiene funcionalidades para almacenar un modelo de proceso de negocio, junto con restricciones variaciones permitidas de ese modelo de proceso de negocio. La información histórica a cerca de las variaciones son realizadas en tiempo de ejecución y también son almacenadas,

de tal forma que pueden ser utilizadas para mejorar el modelo de proceso de negocio. PVR soporta las actividades, el flujo de control, los datos, los recursos, aspectos de monitoreo y autorización y almacena procesos específicos de compañías, instancias de procesos e información histórica acerca de las instancias. Aunque PVR es diseñado para la consulta de variantes de procesos (registros), también provee soporte para consultar definiciones de procesos, además, provee funcionalidades estándar para almacenar y recuperar modelos de procesos y sus variantes. El lenguaje de modelado es propietario y utiliza un lenguaje de consultas de definiciones e instancias, pero no utiliza un lenguaje específico para el almacenamiento y recuperación.

- **Querying Framework** [77] es un entorno para el desarrollo de mecanismos avanzados de búsqueda para un repositorio de procesos de negocio. Este entorno soporta los aspectos de actividades, flujo de control, datos, objetivos, recursos, incluyendo información de estos recursos y autorización. De esta forma, el aspecto de datos es soportado en un alto nivel de abstracción. Además, almacena información acerca de estos aspectos basándose en un repositorio de ontologías RDF, el lenguaje necesario para ingresar esta información es WSML y la información puede ser almacenada en patrones de procesos, fragmentos de procesos, procesos específicos de compañías y, aunque estos no están explícitamente mencionados, referencias de procesos. El repositorio puede ser consultado utilizando expresiones lógicas WSML y no tiene un lenguaje de modelado definido. Es un entorno que no está disponible en línea y es resultado de un proyecto de investigación sin aplicaciones disponibles.
- **Prosero** [78] es un proyecto que combina la gestión de procesos de negocio y la arquitectura orientada a servicios (SOA) que soporta outsourcing de procesos de negocio basado en servicios Web. La semántica de este repositorio consta de 4 componentes: la terminología, el repositorio de modelos de referencia (RMR), el repositorio de servicios Web (WSR) y el repositorio de modelos del Usuario (CMR). Estos componentes pueden ser utilizados para almacenar modelos de procesos de referencia y modelos de procesos específicos de compañías y las instancias de los mismos. Prosero tiene una interfaz externa para herramientas de modelado (estructura organizacional del modelado, modelado de datos y modelado de procesos de negocio) y el motor de ActiveBPEL4People. Los

aspectos soportados por Prosero son: actividad, flujo de control, datos, recursos, autorización y estructura organizacional; además almacena procesos descritos en BPMN y ejecuta procesos BPEL, debido a que tiene un generador de BPEL que puede transformar BPMN en BPEL. No tiene un lenguaje definido para la recuperación de procesos y no se permite realizar consultas semánticas.

- **OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support)** ha sido propuesto para la ejecución de procesos peer-to-peer. Este repositorio se enfoca en el almacenamiento de modelos de procesos de negocio, especificaciones de servicios así como éstas sean provistas o utilizadas por los procesos de negocio e instancias de modelos de procesos de negocio ejecutables. Además, OSIRIS provee funciones típicas peer-to-peer, así como el control de concurrencia y el balance de carga. De igual forma, OSIRIS soporta los aspectos de actividad, flujo de control y datos y almacena modelos de procesos (o servicios) específicos de compañías e instancias de procesos en ejecución. Internamente, los modelos de procesos y las especificaciones de los servicios son almacenados en una base de datos. Utiliza una notación propietaria para el modelado de procesos y tiene funcionalidades básicas para el almacenamiento y recuperación de los modelos de procesos, aunque no especifica el lenguaje empleado.
- **APROMORE** [79] es un repositorio avanzado que mantiene, analiza y reutiliza grandes colecciones de modelos de procesos; además es una plataforma de código abierto, implementado de acuerdo a la arquitectura SOA y se presenta al usuario a través de servicios Web. La representación de los procesos se basa en un formato canónico y utiliza EPC y BPMN como lenguajes de modelado. Actualmente hay una versión de un prototipo disponible en la red y contiene unas funcionalidades básicas: importar y exportar modelos, búsqueda y clasificación de modelos, así como funcionalidades de comparación (búsqueda de similitud de procesos y funcionalidad de gestión de procesos). No proporciona consultas semánticas.

Anexo F

F Colección de Procesos de Negocio de Prueba

Este anexo presenta el conjunto de procesos de negocio de prueba, utilizado para realizar los análisis de calidad y rendimiento sobre el entorno BeMantics presentado en el presente trabajo.

F.1 Procesos de Negocio de Consulta

En la recuperación de procesos de negocio el objetivo es encontrar un conjunto de procesos de negocio del repositorio que sean similares en cuanto a estructura, comportamiento y semántica a un proceso de negocio de referencia ingresado por el usuario. A este proceso de negocio de referencia se lo conoce como proceso de negocio de consulta. A continuación en las tablas Tabla F.1-1 y Tabla F.1-2 se listan los procesos de negocio que se utilizaron como consultas para realizar pruebas de calidad y rendimiento sobre el prototipo BeMantics propuesto en el presente trabajo.

En la Tabla F.1-1 se presenta una descripción y el dominio de aplicación de cada proceso de negocio de consulta. En este caso el dominio puede ser de Telecomunicaciones (Telco) o de Geo procesamiento (Geo).

Procesos de Negocio de Consulta (BP Consulta)			
ID	Nombre de BP	Descripción	Dominio
1	Q_Activate Service	Describe el proceso de activación de un servicio	Telco
2	Q_Best path forecast	Calcula el mejor camino para llegar a algún sitio. Un usuario escoge unas paradas sobre un mapa y el proceso calcula la mejor ruta entre ellos	Geo
3	Q_Bill Delayed Payment	Es el proceso seguido cuando el consumidor de un servicio tiene una factura en mora	Telco
4	Q_Calculate best routes	Calcula el mejor camino para alcanzar un destino. De acuerdo con unos parámetros como coordenadas geográficas, códigos postales, o el nombre; el proceso carga un mapa sobre el cual un usuario puede situar unos puntos de parada, a partir de los cuales se calcula el camino más corto.	Geo
5	Q_Cheking environmental variables	Describe como un usuario consulta las variables ambientales disponibles en una estación. (Clima, presión atmosférica, humedad, radiación solar).	Geo
6	Q_Launching a new product	Proceso para diseñar, generar y promover un nuevo producto.	Telco

Tabla F.1-1: Descripción de los procesos de negocio de consulta y su dominio de aplicación

En la Tabla F.1-2 se presentan las diferentes características de los procesos de negocio de consulta una vez transformados a grafos de acuerdo a sus tipos de nodos, los cuales pueden ser Conectores (compuertas de control de flujo), Eventos (Tareas activadas por acciones externas) y Funciones (Tareas como invocaciones a Servicios Web, Tareas manuales las cuales contienen entradas y salidas). Además se muestra el número de aristas que cada proceso de consulta contiene.

ID	Nombre de BP	Aristas	Conectores	Eventos	Funciones
1	Q_Activate Service	16	4	2	8
2	Q_Best path forecast	9	1	1	6
3	Q_Bill Delayed Payment	16	3	3	6
4	Q_Calculate best routes	10	1	0	7
5	Q_Cheking environmental variables	9	1	6	1
6	Q_Launching a new product	9	1	1	6

Tabla F.1-2: Características de los procesos de negocio de consulta.

F.2 Procesos de Negocio de Comparación

En esta sección se presenta los procesos de negocio de comparación, los cuales son aquellos procesos de negocio almacenados en el repositorio y que constituyen el espacio de búsqueda sobre el cual el prototipo BeMantics encontrará similitudes con cada uno de los procesos de negocio de consulta, presentado en la sección anterior. A continuación en las tablas Tabla F.2-1 y Tabla F.2-2 se listan los procesos de negocio utilizados para los análisis de calidad y rendimiento del prototipo BeMantics.

En la Tabla F.2-1 se presenta una descripción y el dominio de aplicación de cada proceso de negocio del repositorio.

Procesos de Negocio de Comparación (BP Repositorio)			
ID	Nombre de BP	Descripción	Dominio
1	Acquire Service Mobile	Proceso de encendido de un teléfono móvil y su enganche al servicio	Telco
2	Activate Service	Describe el proceso de activación de un servicio	Telco
3	Advice Of Charge	Calcula y envía al usuario la advertencia de recargo por todos los eventos entregados por el proveedor de un servicio que son consumidos	Telco
4	Best path forecast	Calcula el mejor camino para llegar a algún sitio. Un usuario escoge unas paradas sobre un mapa y el proceso calcula la mejor ruta entre ellos	Geo
5	Billing Advice of Charge	Describe el proceso para obtener la factura de un servicio de acuerdo con las preferencias de un usuario.	Telco
6	Billing Process	Describe el proceso de generación de facturas para un usuario teniendo en cuenta los servicios contratados, los servicios adicionales consumidos y el precio de ellos de acuerdo con la ubicación o estrato al que pertenezca	Telco
7	Bill Delayed Payment	Es el proceso seguido cuando el consumidor de un servicio tiene una factura en mora	Telco
8	BroadbandServiceActivation	Describe el proceso para activar un servicio de banda ancha.	Telco
9	Buy intermediate goods process	Describe el proceso para comprar suministros en una empresa	Geo

10	Calculate best routes	Calcula el mejor camino para alcanzar un destino. De acuerdo con unos parámetros como coordenadas geográficas, códigos postales, o el nombre; el proceso carga un mapa sobre el cual un usuario puede situar unos puntos de parada, a partir de los cuales se calcula el camino más corto.	Geo
11	Calculate epidemic risk	Calcula el riesgo de epidemia causado por una enfermedad, teniendo en cuenta los factores que inciden en su propagación, las condiciones geográficas donde se presenta el brote, la información demográfica y la calidad de los servicios de salud existentes.	Geo
12	CalculateGasDispersion Service	Este es un servicio para calcular la dispersión de un gas tóxico emanado por una planta química.	Geo
13	CalculateToxicGasReleases AccidentRisk	Calcula el riesgo de accidentes causados por el esparcimiento de un gas tóxico.	Geo
14	Calling a taxi	Describe el proceso que un call center realiza para enviar un taxi a un usuario.	Geo
15	Cheking environmental variables	Describe como un usuario consulta las variables ambientales disponibles en una estación. (Clima, presión atmosférica, humedad, radiación solar).	Geo
16	Checking an environmental variable	Consulta los valores pasados y actuales para una variable ambiental y predice las futuras.	Geo
17	Create New Service	Describe el proceso de creación de un servicio nuevo, desde su propuesta hasta el establecimiento de una estrategia de mercadeo	Telco
18	Crime predictor	Predice el número y el tipo de crímenes que ocurrirán en un vecindario.	Geo
19	CustomerCreditCheck	Este proceso revisa el estado de un usuario y verifica su existencia o creo una nueva cuenta si no se encuentra en el sistema.	Geo
20	Customer Retention and Loyalty	Proceso para confirmar si los clientes obtienen de lo que desean de un proveedor de servicio. Mide la satisfacción del cliente.	Telco
21	Delayed bill and reactivation	Describe the process followed when a customer have a delayed bill and the service has been suspended	Telco

22	DevelopAndTrainWorkforce	El desarrollo de una fuerza de trabajo ha emergido para describir un relativamente amplio rango de actividades, políticas y programas	Telco
23	Developing and launching process	Describe el desarrollo y lanzamiento de un nuevo producto en el mercado	Telco
24	Develop Technology Strategy	La estrategia tecnológica es la tarea de construir, mantener y explotar los activos tecnológicos de una empresa	Telco
25	Discussion Programing	Es una práctica de gestión que busca mejorar la eficiencia de los procesos de negocio.	Telco
26	DSLActivation	Describe el proceso para activar un servicio DSL.	Telco
27	DSL Provision	Gestiona el aprovisionamiento de DSL.	Telco
28	Enable Service	Describe el proceso para activar un servicio	Telco
29	Enable Service Provisioning	Describe la forma como un proveedor de servicios adquiere un nuevo cliente y le activa un servicio	Telco
30	Environmental change evaluation	Describe el proceso para evaluar el impacto ambiental en un área de estudio específica comparando el cambio de unas variables ambientales a través del tiempo.	Geo
31	environmental impact assessment	Describe el proceso que evalúa el impacto ambiental en un área de estudio, comparando los cambios de unas variables ambientales en un periodo de tiempo.	Geo
32	Establish a new product on the market	Propone, desarrolla, diseña y promueve un nuevo producto.	Telco
33	Examine environmental variables	Examina datos pasados de variable ambientales.	Geo
34	Finding a place process	Describe el proceso para encontrar un lugar mediante su nombre o las coordenadas geográficas.	Geo
35	Finding transportation routes	Encuentra las rutas de transporte (metro, bus, tranvía, etc.) para ir de un lugar a otro.	Geo
36	Find me	Describe el proceso de inscripción al servicio de localización por SMS, y su funcionamiento.	Geo

37	Free area study	Describe el proceso de petición de un área libre para la operar o explotar una mina.	Geo
38	Fulfillment DSL Ordering	Permite a los usuarios pedir un servicio DSL.	Telco
39	Generate Bill	Es el proceo seguido para generar una factura con los consumos recientes de los clientes y teniendo en cuenta facturas anteriores	Telco
40	Global health atlas interactive mapping	Provee una interfaz de usuario amigable que permita a los usuarios desplegar sobre el mapa de un área geográfica deseada, las enfermedades allí presentes, la ubicación de centros médicos, colegios, vías y características geográficas.	Geo
41	Handling customer problems	Describe como un proveedor de servicios maneja los problemas de los usuarios.	Geo
42	Hydrologic analyzer	Describe el proceso para examinar indicadores hidrológicos. (Patrón de flujo, avalanchas o patrón de crecimiento de una rivera, etc.)	Geo
43	ImageCaptureSystem	Proceso de captura de imágenes, su optimización, reproducción y despliegue.	Geo
44	Impact infrastructure of a service network	Edita el trazado de una red teniendo en cuenta el impacto ambiental ocasionado.	Geo
45	Implement a service	Describe el proceso de activación de un servicio.	Geo
46	Inputs buying process	Describe el proceso de compra de insumos en una empresa	Geo
47	Installation service order	Describe el proceso para activar un servicio consultando las variables ambientales para asegurarse de tener buenas condiciones climáticas para la instalación	Telco
48	IntervenirServicioTBC	Proceso de suspensión o reactivación de un servicio teniendo en cuenta la información de cartera	Telco
49	Invoice payment	Describe el proceso para facturar y pagar un servicio	Telco
50	Launching a new product	Proceso para diseñar, generar y promover un nuevo producto.	Telco
51	Locate target users for a service	Proceso para encontrar posibles clientes de acuerdo con las especificaciones de un servicio	Telco

52	Manage Corporate Governance	Es el conjunto de procesos, costumbres, políticas, leyes e instituciones que afectan la forma como una compañía es dirigida, administrada o controlada	Telco
53	Manage Customer Problem	Describe el proceso para atender las quejas de los clientes	Telco
54	Manage Customer QoS SLA	Detecta, administra y reporta el desempeño de QoS/SLA con el cliente.	Telco
55	ManageCustomer Relationships	Evalúa el nivel de satisfacción de los usuarios.	Telco
56	Management indicators	Describe la forma como el personal de una empresa proveedora de telefonía puede acceder a los indicadores de gestión consultando una base de datos SIG para conocer datos acerca del funcionamiento del servicio en diferentes ubicaciones	Geo
57	Manage Positioning	El posicionamiento es un proceso que se centra en transmitir el valor del producto a los compradores, lo que resulta en una familia de documentos que maneja todas las comunicaciones salientes.	Geo
58	Managing a new service	Proceso seguido para evaluar el nivel de aceptación de un servicio nuevo	Geo
59	Market a new product	Analiza, desarrolla y vende un nuevo producto.	Telco
60	Market a new service	Describe como un proveedor de un servicio escoge los mejores lugares para promocionarlo, es decir donde encuentra más clientes potenciales	Telco
61	NewCustomerRegistration	Describe el proceso de registro de un cliente	Telco
62	OnlineOrderDeliveryProcess	Proceso de compra y envío de productos en línea. El proceso revisa la capacidad de pago del cliente y verifica la disponibilidad de los productos.	Telco
63	PasswordRetrieval	Proceso de restitución de una contraseña olvidada. El usuario llena un formulario y después que el sistema revise la información personal, la contraseña es enviada al correo electrónico.	Telco

64	Perform Project Accounting	El uso de la Contabilidad de proyectos proporciona a los gerentes la habilidad para evaluar correctamente y monitorear el presupuesto y asegurarse que el proyecto procede de acuerdo a este. Los gerentes de proyectos pueden localizar rápidamente cualquier costo excesivo y revisar el presupuesto si es necesario.	Telco
65	Portfolio management	Describe el proceso completo desde la generación de una factura hasta su pago.	Telco
66	Pre-Sales	Describe el proceso que tiene lugar antes de vender un servicio.	Telco
67	Present a product	Describe el proceso de mercadeo y presentación de un producto.	Telco
68	Promote New Product	Proceso para proponer, generar y promover un nuevo producto.	Telco
69	Provisioning process	Describe el proceso para proveer un servicio.	Telco
70	Public health viewer	Permite consultar información sobre una enfermedad específica y su relación con determinados asentamientos poblacionales, entorno social, ambiente natural y servicios de salud.	Geo
71	Recruitment process	Describe el proceso de contratación desde la publicación de una vacante hasta la contratación	Telco
72	Register finding devices	Describe el proceso para registrar el servicio que permite encontrar dispositivos electrónicos mediante posicionamiento wifi, torres celulares o GPS.	Geo
72	Register finding devices	Describe el proceso para registrar el servicio que permite encontrar dispositivos electrónicos mediante posicionamiento wifi, torres celulares o GPS.	Geo
74	RequestService	Describe el proceso para solicitar y adquirir un servicio:	Telco
75	Resource Trouble Detection & Resolution	Proceso de seguimiento al desempeño de los recursos y resolución de problemas	Telco

76	searching device	Describe el proceso para encontrar un dispositivo móvil registrado en un servicio de localización de dispositivos (computador portátil o un teléfono móvil), mediante posicionamiento wifi, torres celulares o GPS.	Geo
77	Sell a service	Describe el proceso seguido para vender un servicio.	Telco
78	ServiceActivationProcess	Describe el proceso para activar un servicio.	Telco
79	ServiceDeliveryProcess	Describe las tareas para activar un servicio.	Geo
80	service demand calculator	Permite analizar la densidad de la demanda de un servicio en un área geográfica específica, mostrando detalles acerca de los usuarios y la estructura disponible.	Geo
81	Service Flow	Describe el flujo de un servicio desde que se solicita hasta su facturación y pago.	Geo
82	Service network trace	Proceso para administrar la ubicación, el camino y el estado de construcción de una red de telecomunicaciones (existente, planeada o removida).	Geo
83	ServiceProvisioningProcess	Describe el proceso utilizado para ofrecer un servicio de acuerdo a las capacidades instaladas	Telco
84	Service Retirement Process	Describe el proceso para identificar los servicios existentes que son inviables y su retiro del mercado.	Telco
85	Service sale	Describe la forma como un servicio de telecomunicaciones es solicitado e instalado.	Telco
86	SMS finding place	Describe el proceso para encontrar un lugar deseado cerca de la ubicación en la que se encuentra el usuario, enviando un SMS con la petición (Museo, biblioteca, almacén, estación de tren, cajero automático, etc).	Telco
87	SMS places search engine	Describe el proceso para encontrar un lugar deseado cerca de la ubicación en la que se encuentra el usuario, enviando un SMS con la petición (Museo, biblioteca, almacén, estación de tren, cajero automático, etc).	Telco
88	Solve Customer Problem	Describe el proceso para el manejo de los problemas de los usuarios	Telco

89	Solve Network Service Problem	Describe el proceso de manejo de los problemas de la red de servicios	Telco
90	Solve Telecommunication Problems	Describe como se solucionan los problemas en los procesos de telecomunicaciones, e intenta adquirir las mejores ideas acerca de cómo resolverlos correctamente.	Telco
91	Supply Fulfillment	Es en el sentido más general el proceso completo desde la investigación del punto de venta para entregar el producto al consumidor. A veces el cumplimiento de una orden se usa para describir el acto más estrecho de distribución o de función logística, sin embargo, en el sentido amplio se refiere a la forma como las compañías responden a sus clientes	Telco
92	Suspend service	Describe el proceso para suspender un servicio	Telco
93	Telecom Service Delivery	Describe el proceso de activación de un servicio	Telco
94	Test of the environmental impact	Describe el proceso para evaluar el impacto ambiental en un área de estudio comparando los cambios de una variable a través del tiempo	Geo
95	Vacation planner	Permite planear unas vacaciones escogiendo un destino y algunos planes y sitios de interés. Al final del proceso el usuario obtendrá un mapa con el mejor recorrido que abarque todas las actividades y lugares deseados, con un itinerario que incluya tiempos estimados, presupuestos y hospedaje.	Geo
96	Vendor selection process	Describe el proceso para seleccionar los proveedores de suministros	Geo
97	VPN Setup	Una VPN utiliza redes de telecomunicaciones públicas para conducir la comunicación de datos privados. La mayoría de la implementación de las VPN utilizan la Internet como infraestructura pública y una variedad de protocolos especializados para soportar comunicaciones privadas a través de la Web.	Telco

98	VPN Test	Es el proceso seguido para entregar información a los socios de una red VPN con la información acerca de la calidad del producto o servicio bajo una prueba	Telco
99	WebServiceApproval	Proceso de aprobación de un servicio web, solicita la disponibilidad de un servicio y se envía el mensaje de aceptación o rechazo.	Telco
100	WebServiceRegistry	Proceso de registro de un servicio web. Un servicio es registrado en una página web, luego es confirmado y posteriormente cargado para ser usado.	Telco

Tabla F.2-1: Descripción de los procesos de negocio del repositorio y su dominio de aplicación

En la Tabla F.2-2 se presentan las diferentes características de los procesos de negocio del repositorio una vez transformados a grafos de acuerdo a sus tipos de nodos.

ID	Nombre de BP	Aristas	Conectores	Eventos	Funciones
1	Acquiere Service Mobile	16	3	1	8
2	Activate Service	16	4	2	8
3	Advice Of Charge	8	0	0	7
4	Best path forecast	9	1	1	6
5	Billing Advice of Charge	8	1	0	5
6	Billing Process	12	2	0	7
7	Bill Delayed Payment	16	3	3	6
8	BroadbandServiceActivation	11	2	1	4
9	Buy intermediate goods process	19	3	2	12
10	Calculate best routes	10	1	0	7
11	Calculate epidemic risk	10	2	0	6
12	CalculateGasDispersionService	10	2	0	6
13	CalculateToxicGasReleasesAcccidentRisk	15	4	0	8
14	Calling a taxi	17	3	1	10
15	Cheking environmental variables	9	1	6	1
16	Checking an environmental variable	9	2	0	5
17	Create New Service	14	3	1	8
18	Crime predictor	9	2	1	4

19	CustomerCreditCheck	13	4	0	6
20	Customer Retention and Loyalty	11	3	1	5
21	Delayed bill and reactivation	10	1	2	5
22	DevelopAndTrainWorkforce	27	6	1	16
23	Developing and launching process	11	1	1	8
24	Develop Technology Strategy	8	0	0	7
25	Discussion Programing	20	6	3	6
26	DSLActivation	7	1	2	3
27	DSL Provision	11	2	0	5
28	Enable Service	10	2	2	5
29	Enable Service Provisioning	12	2	2	7
30	Environmental change evaluation	7	1	1	4
31	environmental impact assessment	5	0	0	4
32	Establish a new product on the market	12	3	1	6
33	Examine environmental variables	4	0	0	3
34	Finding a place process	13	1	1	9
35	Finding transportation routes	13	4	2	5
36	Find me	18	4	1	9
37	Free area study	15	3	0	8
38	Fulfillment DSL Ordering	20	4	0	11
39	Generate Bill	6	1	0	3
40	global health atlas interactive mapping	11	2	0	7
41	Handling customer problems	13	2	2	8
42	Hydrologic analyzer	8	1	0	5
43	ImageCaptureSystem	10	2	0	6
44	Impact infrastructure of a service network	14	2	0	9
45	Implement a service	18	4	2	10
46	Inputs buying process	13	2	1	8
47	Installation service order	18	3	2	11
48	IntervenirServicioTBC	9	1	0	6
49	Invoice payment	7	1	0	4
50	Launching a new product	9	1	1	6
51	Locate target users for a service	9	2	0	5
52	Manage Corporate Governance	25	8	0	12
53	Manage Customer Problem	7	1	1	4
54	Manage Customer QoS SLA	11	3	1	5
55	ManageCustomerRelationships	8	1	1	5
56	Management indicators	16	3	1	9
57	Manage Positioning	11	2	0	7

58	Managing a new service	9	1	0	5
59	Market a new product	9	1	0	6
60	Market a new service	5	0	0	4
61	NewCustomerRegistration	7	1	1	4
62	OnlineOrderDeliveryProcess	23	6	1	12
63	PasswordRetrieval	18	4	1	9
64	Perform Project Accounting	28	8	1	11
65	Portfolio management	24	4	2	13
66	Pre-Sales	16	1	4	8
67	Present a product	4	0	0	3
68	Promote New Product	7	1	1	4
69	Provisioning process	8	1	1	5
70	public health viewer	17	0	0	16
71	Recruitment process	12	1	0	9
72	register finding devices	17	3	1	10
73	register finding devices	17	3	1	10
74	RequestService	21	5	2	10
75	Resource Trouble Detection & Resolution	14	3	1	8
76	searching device	17	4	2	8
77	Sell a service	9	1	1	6
78	ServiceActivationProcess	10	2	1	5
79	ServiceDeliveryProcess	10	2	1	5
80	service demand calculator	11	1	0	8
81	Service Flow	13	2	0	8
82	service network trace	9	1	0	6
83	ServiceProvisioningProcess	13	2	1	8
84	Service Retirement Process	14	4	0	7
85	service sale	14	2	0	9
86	SMS finding place	15	2	2	9
87	SMS places search engine	8	1	1	5
88	Solve Customer Problem	10	2	0	6
89	Solve Network Service Problem	14	3	1	7
90	SolveTelecommunicationProblems	12	2	0	8
91	Supply Fulfillment	19	5	1	8
92	Suspend service	15	3	0	9
93	Telecom Service Delivery	11	2	1	6
94	Test of the environmental impact	9	1	0	6
95	Vacation planner	11	2	0	7
96	Vendor selection process	8	1	1	5

97	VPN Setup	16	4	0	9
98	VPN Test	15	4	0	7
99	WebServiceApproval	13	4	2	5
100	WebServiceRegistry	10	1	1	7

Tabla F.2-2: Características de los procesos de negocio del repositorio.

Anexo G

G Plataforma para Evaluar Sistemas de Recuperación de Procesos de Negocio

G.1 Introducción

En este anexo se presenta la plataforma Web, “*Pertinence Evaluation Tool*”, que permitió a un grupo de jueces humanos emitir juicios de relevancia en un conjunto de 100 procesos de negocio de los dominios de las telecomunicaciones y geo-procesamiento de acuerdo a un conjunto de 6 procesos de negocio de referencia. En estas evaluaciones los BP detectados por los jueces como similares a un BP de consulta, fueron considerados como BP relevantes. De esta manera este modelo es útil para catalogar a una herramienta automática como efectiva, si sus resultados coinciden con los resultados de las evaluaciones de semejanza realizadas por los jueces en la plataforma Web.

Adicionalmente, la herramienta provee un módulo de gestión de procesos, donde se pueden generar y editar el conjunto de procesos de negocio de prueba. Y finalmente, proporciona una visualización de resultados que despliega una clasificación ordenada de procesos relevantes para una consulta determinada, a partir de los resultados de las evaluaciones manuales.

A lo largo de este anexo se describirán las funcionalidades de la herramienta, se expondrá y explicará su arquitectura, y finalmente se analizarán algunas de las interfaces de usuario más importantes.

G.2 Descripción de la Plataforma

En esta sección, se presentan los tipos de usuarios que soporta la plataforma, se describen y explican las funcionalidades principales de la herramienta, teniendo en cuenta quienes tienen acceso a ellas.

La plataforma soporta dos tipos de usuario, el *Administrador* y un usuario *Comparador*, que se describen a continuación:

- *Administrador*: Es el usuario que tienen acceso a la información prioritaria de la aplicación. Dentro de sus funciones se encuentra la de gestionar los procesos de la colección que se carga en la plataforma. Además puede hacer gestión de todos los usuarios, es decir, que puede crear un usuario de cualquier tipo, editar su información y borrarlos. Adicionalmente, puede revisar los resultados de las comparaciones para cada proceso de consulta. Sin embargo, este tipo de usuario no tiene acceso a la realización de las comparaciones.
- *Comparador*: Es el encargado de realizar las evaluaciones manuales de los procesos. Es decir, es quien interactúa directamente con el módulo de comparación. Una persona puede registrarse como comparador, pero no tendrá acceso a los derechos del administrador. Es decir, que un usuario de este tipo no podrá gestionar usuarios, ni gestionar procesos, y tampoco podrá consultar los resultados.

Cuando un usuario de este tipo se registra desarrolla una encuesta para medir su nivel de conocimiento acerca de servicios y procesos de negocio, workflows, BPMO (Business Process Model Ontology); y su dominio del

idioma Inglés. De acuerdo con los resultados, el *Comparador* se catalogará dentro de uno de tres tipos:

- Experto: Es quien alcanza más de un 60% en su nivel de conocimiento.
- Intermedio: Será el usuario cuyo conocimiento se encuentre dentro del 30% y el 60%.
- Novato: Será quien su conocimiento no alcance el 30%.

Una vez registrados, los usuarios tienen acceso a las diferentes funcionalidades que ofrece la plataforma, las cuales se encuentran distribuidas dentro de los módulos que se describen en la arquitectura.

G.3 Diagrama en Capas

La herramienta está dividida en tres capas: La de aplicación que provee las interfaces de usuario y las funcionalidades básicas de la plataforma; la capa de mediación donde se encuentran todas las interfaces de programación de aplicaciones (API's) utilizadas en el desarrollo de la herramienta; y finalmente la capa de fundamentación que incluye el software básico que se requiere para su funcionamiento (Figura G.3-1).

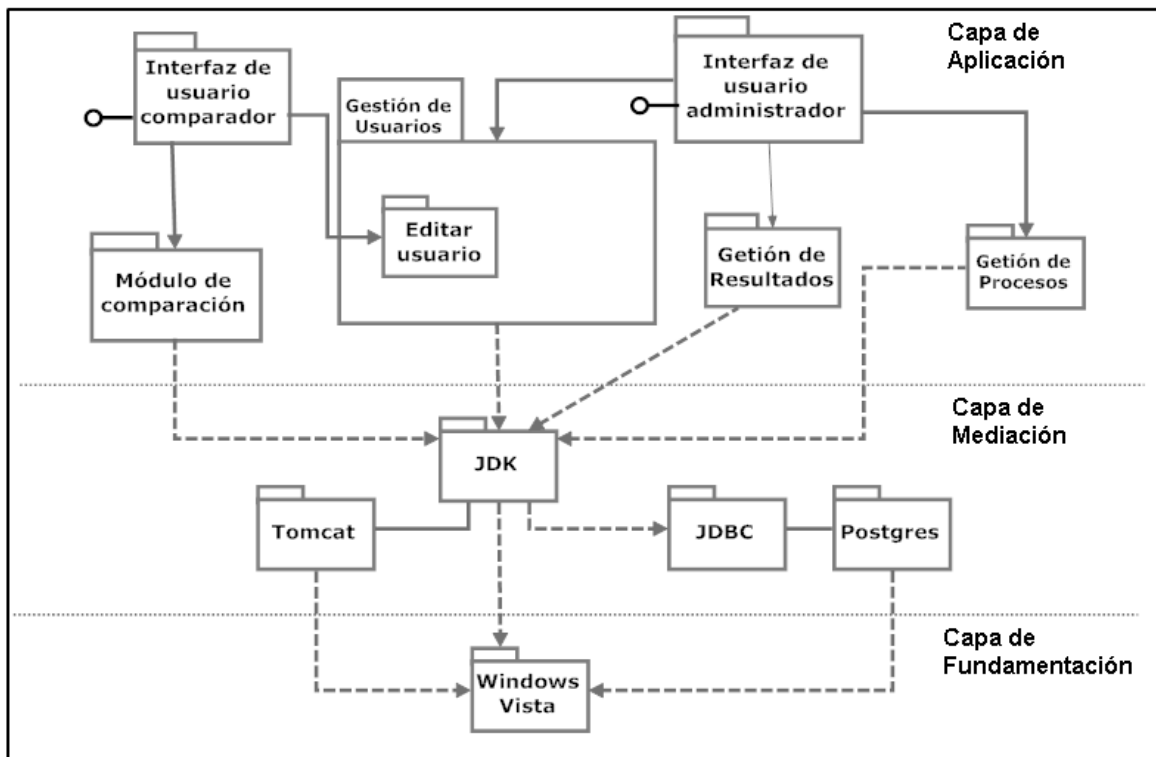


Figura G.3-1: Diagrama en Capas para la Plataforma para Evaluar Sistemas de Recuperación de Procesos de Negocio

A continuación se describe cada una de las capas de la arquitectura para la plataforma de evaluación de pertinencia de la recuperación de procesos de negocio.

G.3.1 Capa de aplicación

Interfaz de usuario administrador: Proporciona al administrador el acceso a todas las funcionalidades de las que es responsable. Da acceso a las gestiones de usuarios, de resultados y de procesos.

- *Interfaz de usuario comparador:* Provee a los usuarios las interfaces y el acceso a la funcionalidad, para realizar las comparaciones manuales entre procesos.
- *Módulo de gestión de usuarios:* Este módulo permite agregar, editar y borrar usuarios en la plataforma y se comunica con la base de datos relacional del

módulo de almacenamiento, para editar las tablas relacionadas con los usuarios. Sus principales funciones son:

- *Agregar Usuario*: Un *Administrador* de la plataforma puede agregar cualquier tipo de usuario, sin embargo es el único que posee los permisos para agregar usuarios con su mismo rol. No obstante, en la página principal de la plataforma existe el link “*Register*” que permite a cualquier persona interesada en participar de las comparaciones, hacer un registro como *Comparador* (Figura G.3-2)

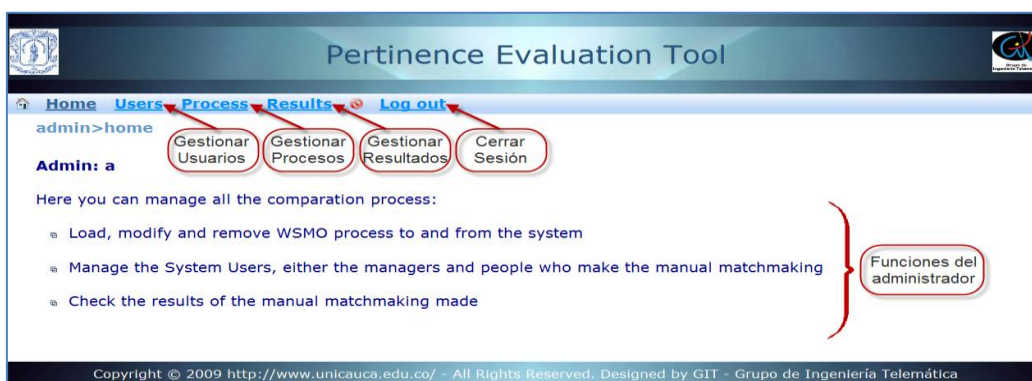


Figura G.3-2: Página inicial para el Administrador

- *Editar Usuario*: El administrador tiene acceso a toda la información de los usuarios desde su menú de gestión de usuarios y puede modificar la información relacionada con cualquiera de ellos. Por otra parte, una vez un *Comparador* ha iniciado su sesión, puede modificar la información que suministró en su registro inicial.
- *Eliminar Usuario*: El derecho para borrar un usuario de la plataforma es exclusivo del *Administrador*. Desde la página de gestión de usuarios, puede eliminar usuarios de cualquier tipo y su información relacionada.
- *Módulo de gestión de procesos*: La gestión de los procesos permite administrar los procesos de la colección WSML del Módulo de Almacenamiento, es decir, permite crear los procesos, especificar los que

servirán de consulta, visualizarlos y eliminarlos. A las funciones que provee este módulo, sólo tienen acceso los Administradores de la plataforma. La gestión de los procesos incluye:

- Agregar Procesos Desde la interfaz para la gestión de los procesos (
- Figura G.3-3) se puede acceder a la adición de Procesos a la colección. Para esto, el *Administrador* debe suministrar los archivos que describen el proceso a ingresar (.wsml y .wsml.layout).

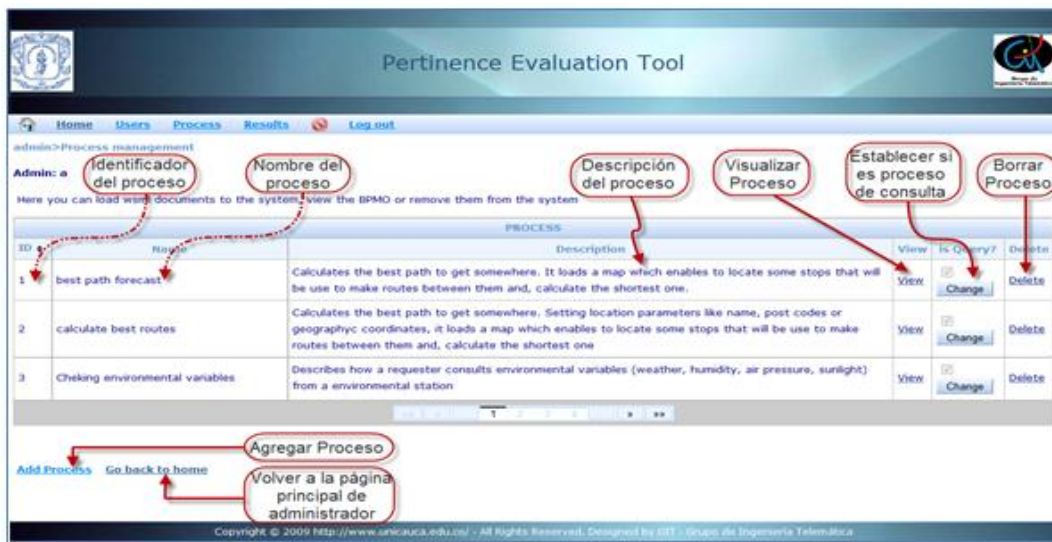


Figura G.3-3: Interfaz para gestionar Procesos

- Visualizar Procesos: Mediante esta función el *Administrador* tiene acceso a toda la información relacionada con un proceso, es decir, puede ver el nombre, la descripción en lenguaje natural del funcionamiento del proceso, la lista de las actividades que lo componen con sus descripciones, parámetros de entrada y salida, la estructura gráfica (BPMO), y si el proceso es de consulta o no (Figura G.3-4).

Pertinence Evaluation Tool

Home Users Process Results Log out

admin>View Process

Process: vacation planner

ID: 0

Description
Helps to plan holidays by choosing a destination and some activities and places of interest. At the end the user will obtain a map with the best course to visit the desired places/activities and an itinerary with the estimated times and money investments in each place/activity and, different choices for accommodation

Activities

Activity Name	Activity Description	Activity Inputs	Activity Outputs
show course map and itinerary	Shows to the user a map with the best course, the desired places/activities, distances and estimated times; also, shows a table with a suggested itinerary, and a estimation of the budget	[best_course_map, accommodation_list, total_distance, estimated_times, estimated_budget]	[estimated_budget, best_course_map, itinery]
choose desired places/activities	Allows the user to choose the places that wants to visit, and the activities that wants to make. Also, the user specifies the departure place to make the calculation of distances, routes and times	[touristic_activities_list, touristic_places_list]	[departure_place, choosed_activities_list, choosed_places_list]
search accommodation	It receives the information of times and places and makes an itinerary where it can suppose how many days the holidays will take and where the user will need accommodation. It looks for some accommodation options and suggest them	[best_course_estimated_time, total_distance, visits_activities_estimated_time, best_course_map, estimated_money]	[accommodation_list, estimated_times, estimated_budget, best_course_map, total_distance]
search for touristic places/activities	Looks for places (museums, cities, monuments, parks, churches, etc) and activities (hiking, extreme sports, horse rides, etc) in the specified destination and sends to the user a list with them	[destination_name]	[touristic_activities_list, touristic_places_list]
calculate best course	It looks for the geographic information of the places and activities desired, locate them in a map, calculates the best course showing distances and estimated times by car.	[choosed_activities_list, departure_place, choosed_places_list]	[total_distance, best_course_map, best_course_estimated_time]
set holiday destination	Allows the user to specify a region or a city that wants to visit	[none]	[destination_name]
estimate time and money investments	It looks for the prices and times to visit each place or to make each activity and estimates the investment in time and money	[choosed_activities_list, choosed_places_list]	[visits_activities_estimated_time, estimated_money]

Structure

```

graph TD
    Start((Start)) --> SetDest[set holiday destination]
    SetDest --> SearchTour[search for touristic places/activities]
    SearchTour --> ChoosePlac[choose desired places/activities]
    ChoosePlac --> ParallelSplit{ }
    ParallelSplit --> CalcBest[calculate best course]
    ParallelSplit --> EstTime[estimate time and money investments]
    CalcBest --> SimpleMerge{ }
    EstTime --> SimpleMerge
    SimpleMerge --> SearchAccom[search accommodation]
    SearchAccom --> ShowMap[show course map and itinerary]
    ShowMap --> End((End))
  
```

Is Query Process?

[back to Process](#)

Copyright © 2009 http://www.unicauca.edu.co/ - All Rights Reserved. Designed by G11 - Grupo de Ingeniería Telemática

Annotations:

- Nombre del Proceso
- Identificador del Proceso
- Descripción del Proceso
- Actividades del Proceso con sus descripciones, entradas y salidas
- Estructura gráfica del
- Indicada si el proceso es o no de consulta
- Regresar a la interfaz de gestión de Procesos

Figura G.3-4: Interfaz de visualización de procesos

- **Eliminar Procesos:** Desde la página de gestión de procesos, el *Administrador* puede eliminar un proceso, borrando de la base de datos toda su información relacionada. No obstante, los archivos (wsml y wsml.layout) se conservan en la carpeta donde se almacenan los procesos de la colección.
- **Módulo de comparación:** A éste módulo sólo tienen acceso los *Comparadores*, y es el que permite hacer las evaluaciones manuales de los procesos (Figura G.3-5).

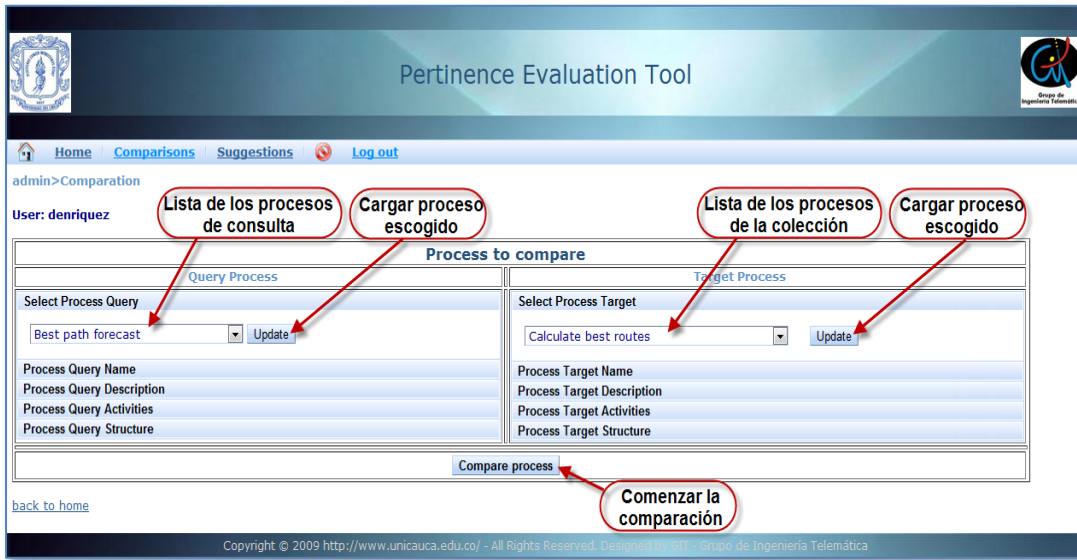


Figura G.3-5: Interfaz principal para la comparación de procesos

- **Módulo de Resultados:** Es el módulo que permite obtener y visualizar la clasificación de los procesos relevantes para una consulta según las evaluaciones proporcionadas por los *Comparadores*. A las funciones que provee este módulo sólo tienen acceso los *Administradores* (Figura G.3-6).

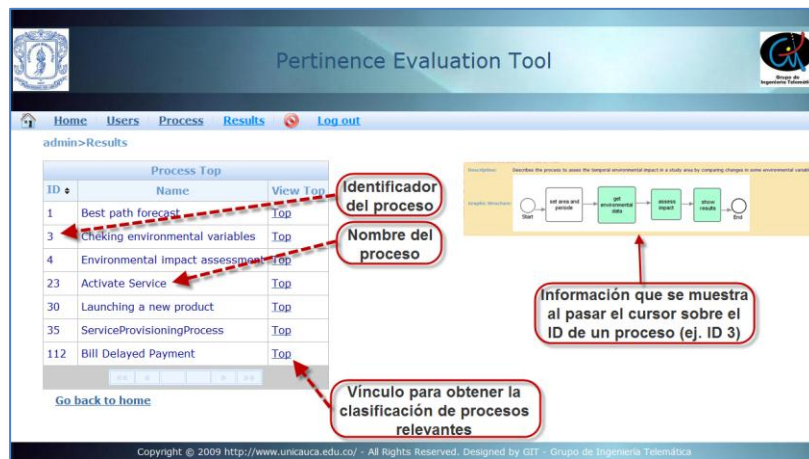


Figura G.3-6: Interfaz principal de los resultados

G.3.2 Capa de mediación

- *Tomcat*: Implementa los JSF (Java Server Faces) proporcionando un ambiente de ejecución en cooperación con el servidor web, para el código Java .
- *JDK (Java Development Kit)*: Es un ambiente de desarrollo integrado (IDE) para escribir aplicaciones Java. Se compone por un ambiente de ejecución que se encuentra en la capa superior del sistema operativo, y permite compilar, depurar y ejecutar la herramienta descrita en este capítulo.
- *JDBC*: Es un API para Java que define la forma como un usuario debe acceder a la base de datos, y provee métodos para consultarla y editarla. Esta interfaz es utilizada por los paquetes de la capa de aplicación que implementan la lógica de la aplicación, para gestionar los datos dentro de la base de datos Postgres.
- *Postgres*: Es el sistema para la administración de bases de datos utilizado para almacenar la información de los usuarios, y los datos recopilados de las evaluaciones manuales.

G.3.3 Capa de fundamentación:

- *Windows Vista Home Premium*: Es el sistema operativo sobre el cual se ejecutaron las aplicaciones necesarias para el funcionamiento de la plataforma de evaluación de la relevancia de procesos de negocio.

Anexo G

H BeMantics: Una Plataforma Recuperar Procesos de Negocio basada en Semántica del Comportamiento

H.1 Introducción

En este anexo describe la plataforma de recuperación de procesos de negocio basada en semántica del comportamiento denominada BeMantics. Esta herramienta se desarrolló con base en el modelo de descubrimiento descrito a lo largo de este trabajo de grado.

H.1.1 Diagrama de Paquetes del Analizador Estructural.

Al igual que el repositorio la arquitectura lógica del repositorio está implementada en 3 capas diferentes (Aplicación, Mediación, Fundamentación), las cuales se presentan en la Figura H.1-1 (a, b y c).

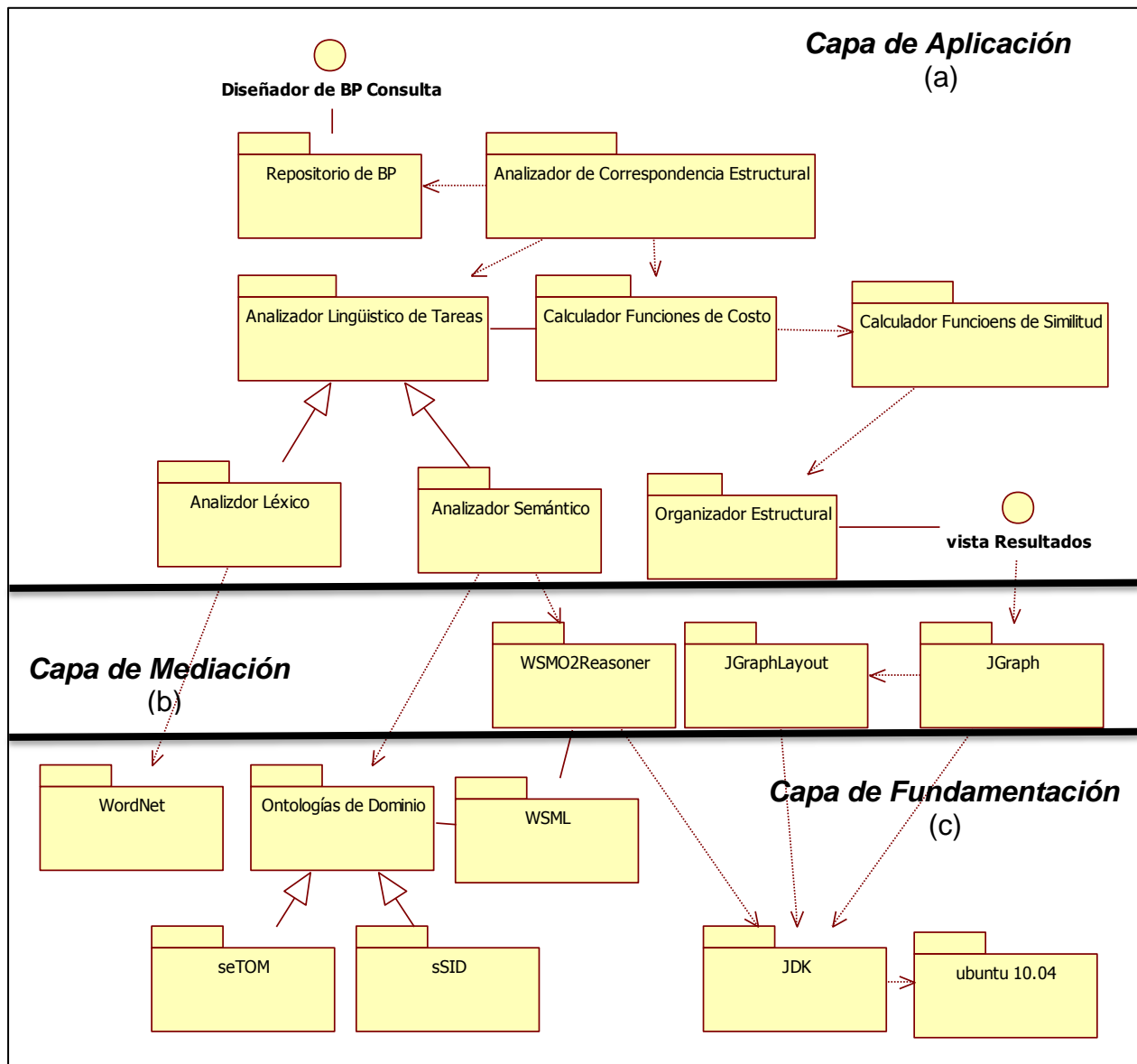


Figura H.1-1. Diagrama de paquetes del analizador Estructural

Capa de Aplicación:

- **Diseñador de BP Consulta:** permite diseñar gráficamente un BP utilizando el lenguaje de modelado BP MO. Para el presente proyecto se utilizó el entorno WSMO Studio (Figura H.1-2)

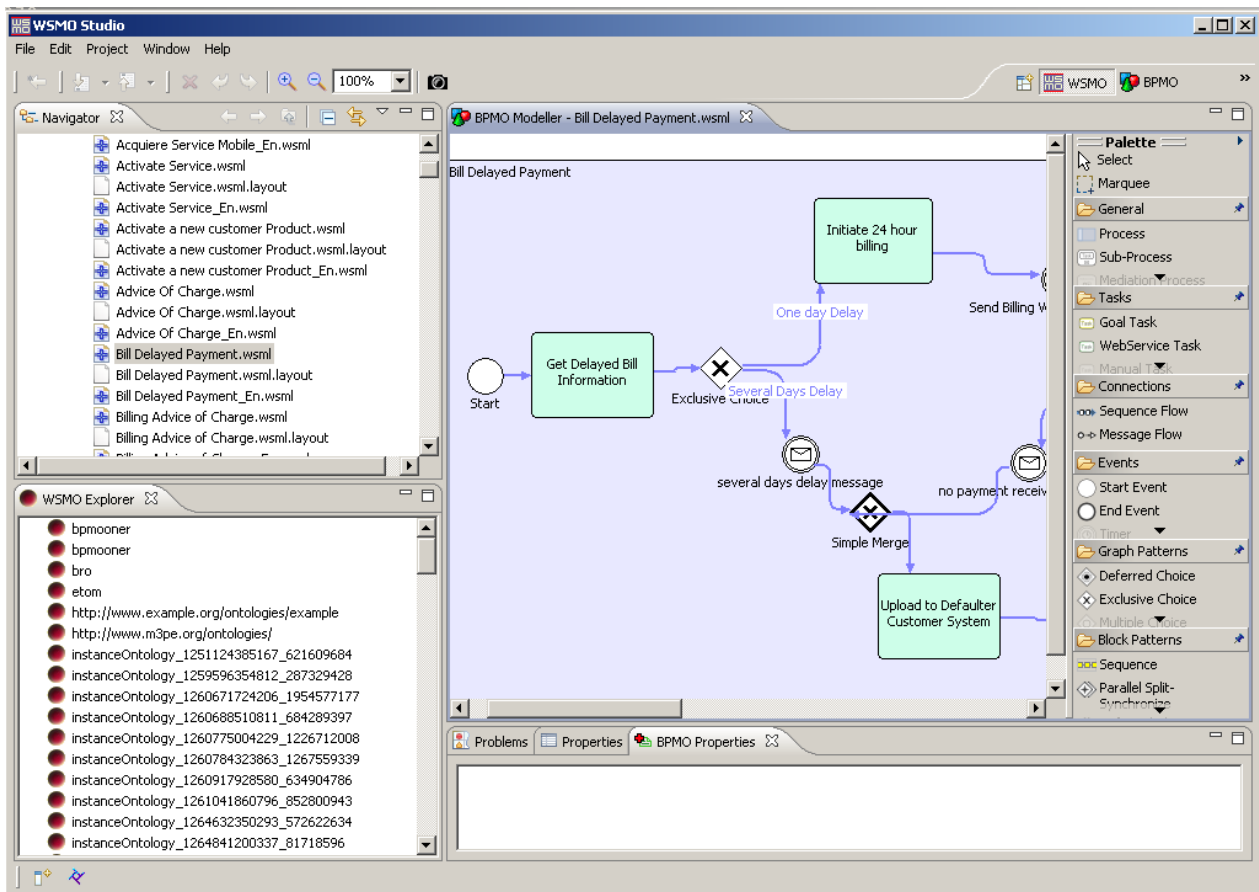


Figura H.1-2: Diseñador de Procesos de Negocio WSMO Studio

- Repositorio de BP:** este paquete corresponde al módulo anterior, presentado en la sección **¡Error! No se encuentra el origen de la referencia.** El cual recibe el proceso BPMO lo transforma a proceso TD y retorna una lista de BP repositorio que cumplen con la condición de contener un conjunto de patrones similar al conjunto de patrones detectados en un BP consulta. En la Figura H.1-3 se presenta un pantallazo de la versión Web del repositorio, en este caso llamado BPMOONER.

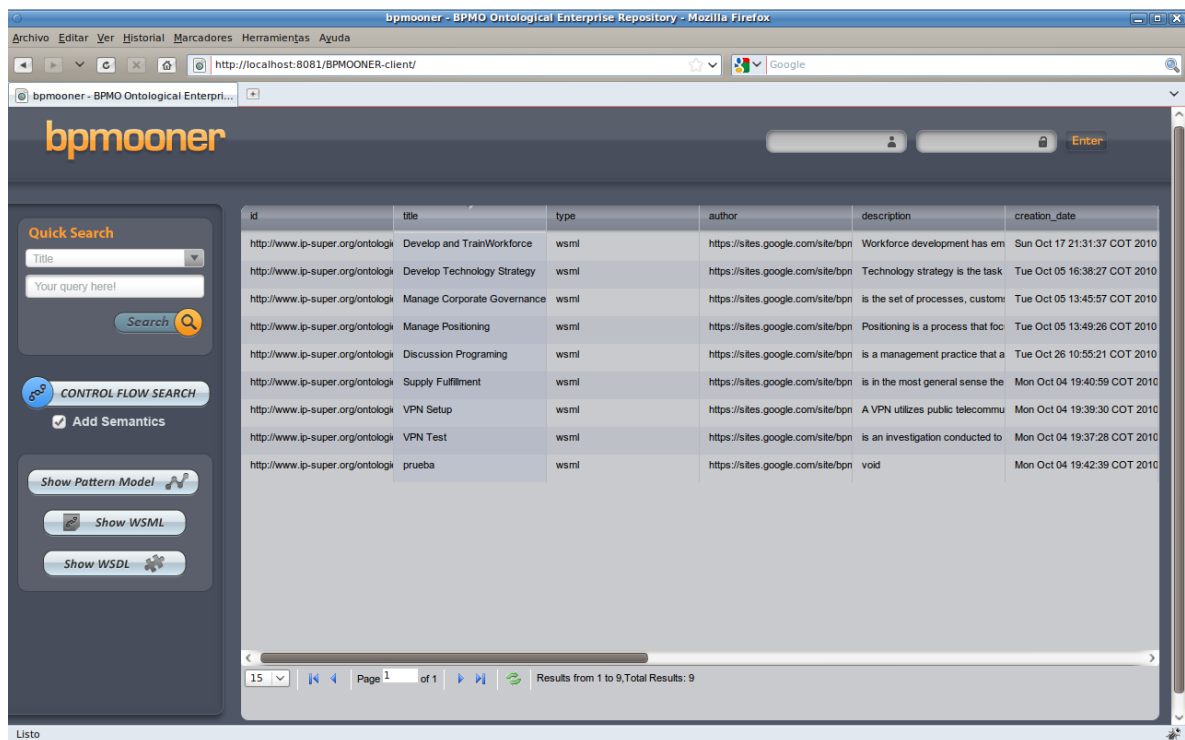


Figura H.1-3: Vista de la versión Web del Repositorio

- **Analizador de Correspondencia estructural:** es el paquete que contiene las clases para ejecutar la correspondencia estructural y semántica entre un BP consulta y un conjunto de BP repositorio a través de un algoritmo de isomorfismo de sub-grafos con corrección de error. .
- **Analizador lingüístico:** es el módulo encargado de determinar la similitud léxica o semántica (según sea el caso, si la tarea tiene o no enriquecimiento semántico) entre dos tareas de tipo función o evento.
- **Analizador Léxico:** es el encargado de encontrar una distancia léxica entre los nombres de una tarea de tipo evento o de una tarea tipo función sin enriquecimiento semántico.
- **Analizador Semántico:** encargado de encontrar una similitud semántica entre dos tareas de tipo función las cuales contienen enriquecimiento semántico tanto en su identificador (nombre) y sus interfaces (entradas/salidas).
- **Calculador de Funciones de Costo:** este método permite calcular la distancia entre dos BP, estimando un costo total de las operaciones de edición ejecutadas por el analizador de correspondencia estructural.

- **Calculador de Funciones de Similitud:** calcula las funciones de similitud a partir de las funciones de costo.
- **Organizador Estructural:** toma los valores de las funciones de similitud y organiza los BP repositorio de mayor a menor similitud.
- **Vista de Resultados:** permite al usuario final la visualización de los resultados de correspondencia entre el BP consulta y cada uno de los BP repositorio, además de las operaciones de edición ejecutadas (Figura H.1-4).

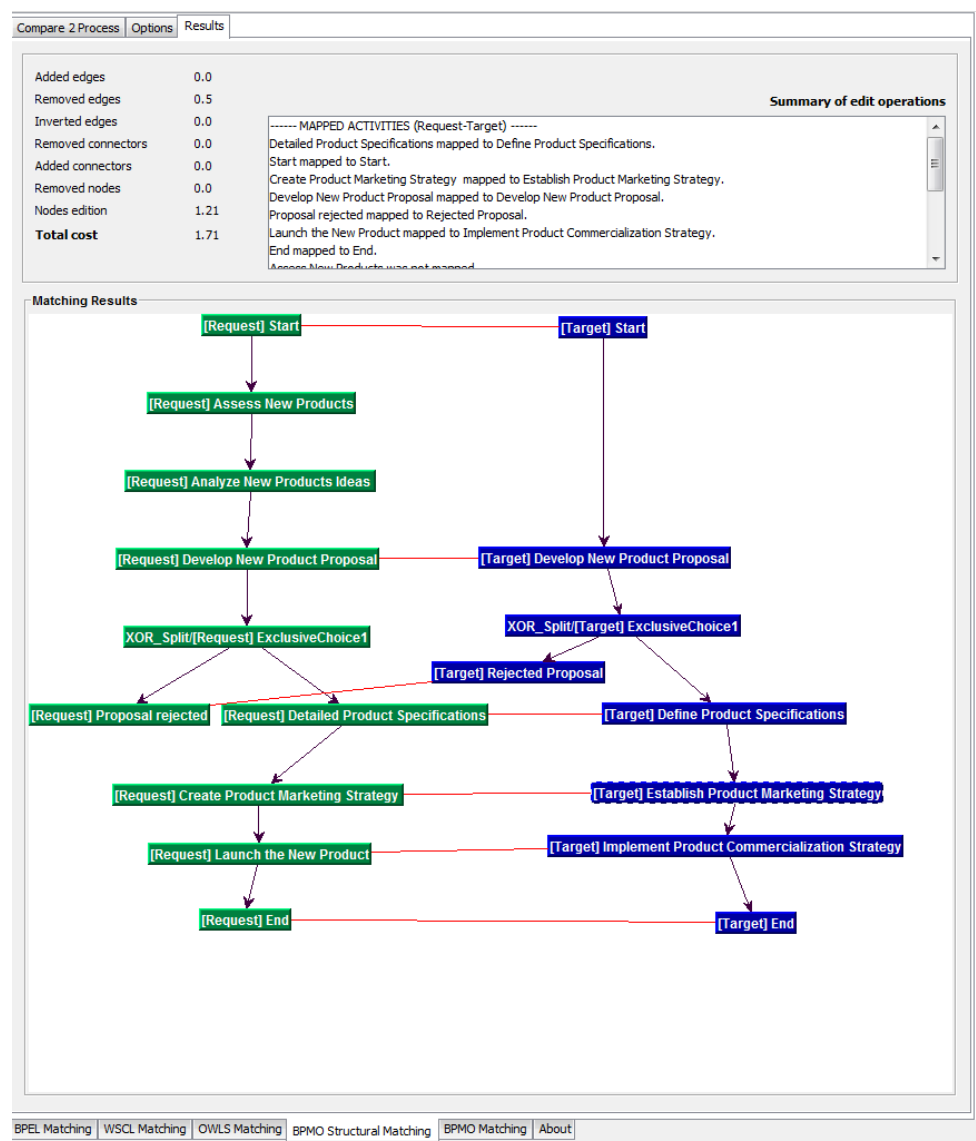


Figura H.1-4: Ejemplo de un resultado del proceso de correspondencia estructural y semántica

Capa de Mediación:

- **WSMO2Reasoner:** es un conjunto de librerías que calculan las diferentes relaciones que existen entre los conceptos de una ontología descrita con el lenguaje WSML.
- **JGraph:** es una librería que permite dibujar grafos bajo el lenguaje Java.
- **JGraphLayout:** permite posicionar los grafos dibujados con JGraphLayout en una interfaz gráfica.

Capa de Fundamentación:

- **WordNet:** es una base de datos léxica que contiene relaciones léxicas entre cadenas de texto.
- **Ontologías de Dominio:** corresponden a ontologías del dominio de las telecomunicaciones las cuales contienen un conjunto de conceptos relacionados que permiten realizar inferencias semánticas. Las ontologías de dominio utilizadas en este trabajo de grado corresponden a seTOM (para los identificadores de las tareas) y sSID (para las entradas y salidas).
- **WSML:** es un lenguaje de modelado semántico, el cual permite modelar servicios web, procesos de negocio y ontologías.
- **Ubuntu 10.04:** es el sistema operativo que soporta al entorno BeMantics.
- **JDK:** corresponde al entorno de desarrollo de aplicaciones Java

Anexo I

I Resultados de los Análisis de Rendimiento y Calidad

I.1 Introducción

En este anexo se presentan los principales resultados obtenidos de los análisis de rendimiento y calidad

Este anexo resume el proceso de adquisición de los resultados de las comparaciones, y los valores obtenidos de la aplicación de las medidas calidad de recuperación y de rendimiento de los dos módulos principales del prototipo BeMantics: el repositorio basado en semántica del comportamiento y el analizador estructural y semántico. El plan de pruebas propuesto contiene dos tipos de análisis, el primero que evalúa el rendimiento de la ejecución y el segundo la calidad de los resultados.

I.2 Análisis de Rendimiento

En este análisis se prueban los tiempos de ejecución tanto del repositorio como del analizador estructural y semántico. Las características del servidor de prueba sobre el cual se ejecutaron los análisis se presentan en la Tabla I.2-1.

Característica	Valor
RAM	4 GB
Procesador	Intel i3-530 (2.93 GHz)
Sistema Operativo	Linux Ubuntu 9.10
Lenguaje de Programación	J2SDK 1.6
Entorno de Desarrollo	Netbeans 6.9
Gestor de Bases de Datos	PostgreSQL 8.4

Tabla I.2-1: Características del servidor de prueba

I.2.1 Análisis de Rendimiento del Repositorio

El análisis de rendimiento del repositorio se hace en las dos fases que éste tiene en su ejecución, la fase de almacenamiento y la fase de recuperación o generación de la lista ordenada de procesos con pre-correspondencia.

Almacenamiento: en esta subsección se presentan los análisis de rendimiento de la fase de almacenamiento del repositorio.

Nodos	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	22	25
Tiempo (ms)	18,2	18,2	19,5	18,9	18,7	18,5	18,5	19,1	17,9	19,5	18,1	17,7	20,5	17,3	18,6	19,4	19,7	21,1

Tabla I.2-2: Tiempo de almacenamiento vs Número de nodos de los procesos de negocio.

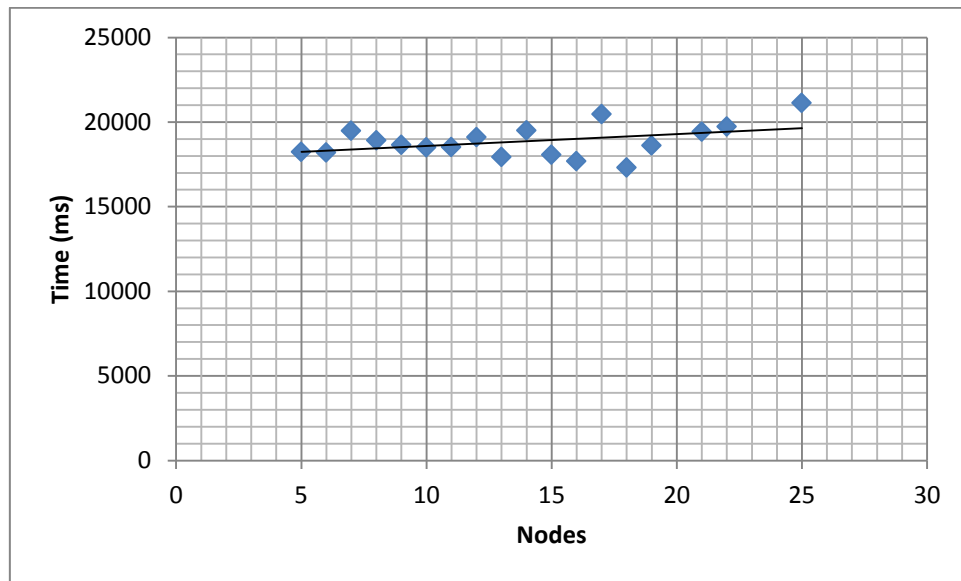


Figura I.2-1: Tiempo de almacenamiento vs Número de nodos (Nodes) de los procesos de negocio.

Aristas	Tiempo (ms)	Aristas	Tiempo
4	18,2	15	17,4
5	17,4	16	19,5
6	19,8	17	15,8
7	19,1	18	18
8	18,6	19	18,3
9	18,4	20	20,5
10	18,4	21	20,9
11	18,7	23	19,4
12	19	24	19,7
13	18,3	25	19,7
14	19,2	27	21,1
15	17,4	28	19,7

Tabla I.2-3: Tiempo de Almacenamiento vs Número de Aristas (Edges) de los procesos de negocio

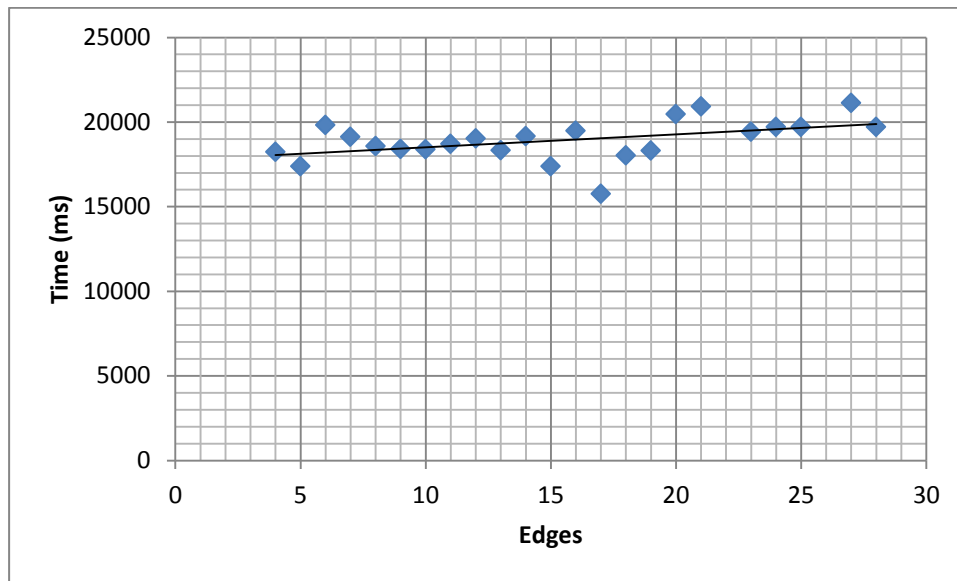


Figura I.2-2: Tiempo de Almacenamiento vs Número de Aristas de los procesos de negocio

Recuperación: en esta subsección se presentan los resultados respectivos a la recuperación de procesos de negocio en el repositorio, es decir la fase de clasificación de resultados de pre-correspondencia basada en semántica del comportamiento.

Consultas	Tiempo (ms)	Procesos de Negocio
Q1	36,41	90
Q2	34,3	90
Q3	30,63	85
Q4	35,66	95
Q5	10,95	37
Q6	17,73	59

Tabla I.2-4: Tiempo de recuperación y número de procesos de negocio clasificados por cada consulta

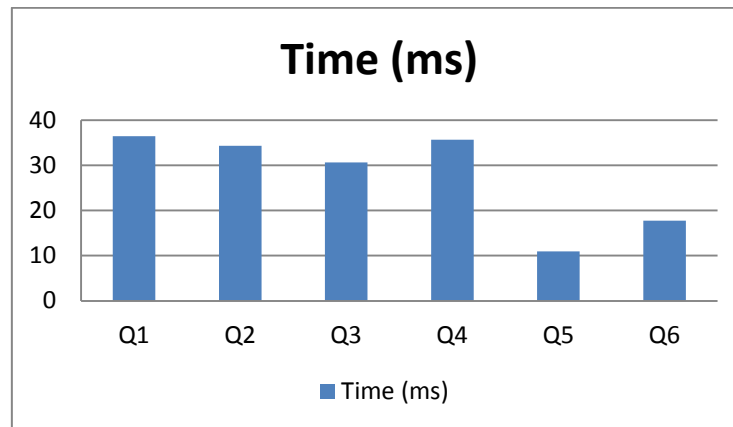


Figura I.2-3: Tiempo de recuperación por cada consulta

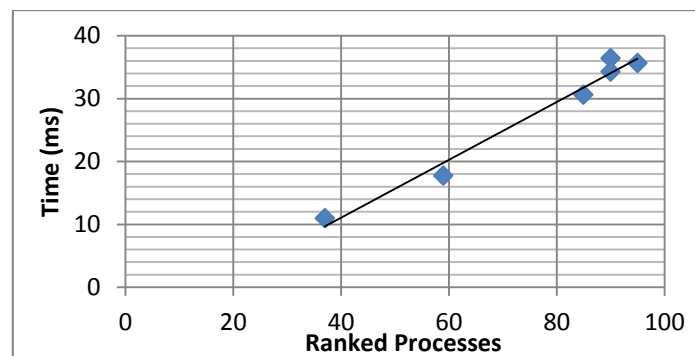


Figura I.2-4: Número de procesos de negocio clasificados vs tiempo de recuperación

I.2.2 Análisis de Rendimiento del Analizador Estructural y Semántico

Aristas	Tiempo (ms)	Aristas	Tiempo (ms)
4	10196,8333	16	955125,72
5	37449,1667	17	755784,92
7	124114,704	18	426650,063
8	210596,079	19	718862,6
9	299679,217	20	931887,3
10	454008,904	21	606242,412
11	459696,933	22	1755644
12	484720,04	23	835694,6
13	677414,148	24	618246,444
14	581587,711	25	631413,25
15	625061,789	27	1513637,5

Tabla I.2-5: Tiempo de correspondencia vs número de aristas

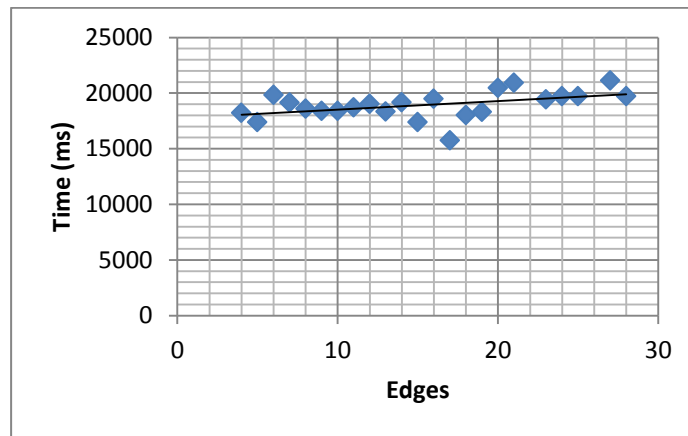


Figura I.2-5: Tiempo de correspondencia vs número de aristas

Tiempo (ms)	Nodos	Tiempo (ms)	Nodos
10196,8333	5	697496,4	16
37449,1667	6	713300,32	17
118808,41	8	692954	18
260735,676	9	736416,375	19
358738,628	10	936376	20
441299,703	11	1050174,4	21
533444,483	12	450795,65	22
633700,419	13	603609,8	23
748249,705	14	1513637,5	25
594323,632	15		

Tabla I.2-6: Tiempo de correspondencia vs número de nodos

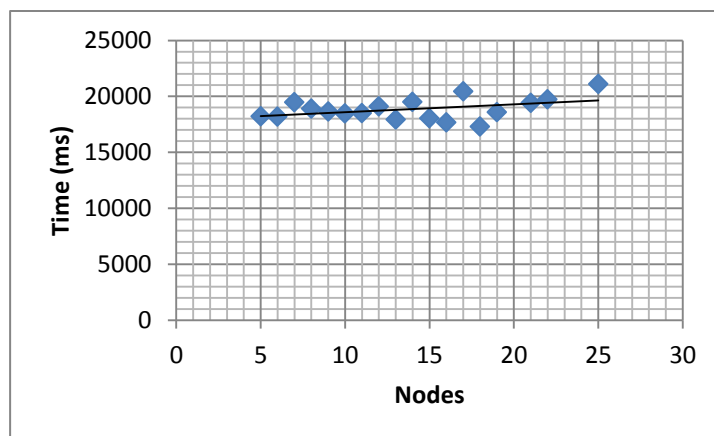


Figura I.2-6: Tiempo de correspondencia vs número de nodos

I.3 Análisis de Calidad

El análisis de calidad evalúa la pertinencia de los resultados obtenidos por el prototipo BeMantics (Evaluación Automática) respecto a unos resultados relevantes obtenidos por los jueces humanos (Evaluación Manual) a través de la plataforma “Pertinence Evaluation Tool” (Anexo G).

En la evaluación manual se obtuvieron 3500 comparaciones provenientes de 5 jueces que evaluaron 6 consultas frente a los 100 elementos de la colección de prueba. La Tabla I.3-1 lista los procesos de negocio utilizados como consulta y el anexo F presenta la lista total de procesos de negocio del repositorio y los procesos de negocio de consulta.

ID consulta	Nombre del Proceso
Q1	Best path forecast
Q2	Checking environmental variable
Q3	Environmental impact assessment
Q4	Activate service
Q5	Launching a new product
Q6	Bill delayed payment

Tabla I.3-1: Procesos de negocio de consulta

Para la evaluación manual los criterios que se tuvieron en cuenta se muestran en la Tabla I.3-2.

Criterios	
CF	Control de flujo
DC	Dependencia Causal
EG	Estructura gráfica
NA	Nombre de las actividades
DA	Descripción de las actividades
EA	Entradas de la actividades
SA	Salidas de las actividades

Tabla I.3-2: Criterios analizados en las evaluaciones manuales

Por otra parte, en la evaluación automática los criterios que el prototipo BeMantics evalúa en los procesos de negocio son la estructura, la lingüística y el comportamiento secuencial. La Tabla I.3-3 muestra la relación entre los criterios de la evaluación manual y los criterios evaluados por la herramienta BeMantics (evaluación automática).

Análisis	Evaluación Automática	Evaluación Manual						
		CF	DC	EG	NA	DA	EA	SA
A	Estructura	20%	20%	20%	10%	10%	10%	10%
B	Lingüística				25%	25%	25%	25%
C	Comportamiento Secuencial	100%						

Tabla I.3-3: Criterios evaluación Manual vs Criterios evaluación Automática

La evaluación de la pertinencia se realizó utilizando medidas de efectividad como la *precisión* (P), que evalúa la habilidad del sistema para recuperar solamente elementos relevantes de acuerdo a una consulta específica [80]; y la *exhaustividad* (R), que mide la capacidad del sistema para obtener todos esos elementos relevantes. Estas medidas se puede clasificar en binarias o gradadas: las medidas de efectividad binarias (P y R) permiten relacionar el número de elementos relevantes recuperados ($BPr \cap BPe$) con la cantidad de elementos recuperados (BPe) y los relevantes (BPr), como se muestra en las ecuaciones 1 y 2 [81].

$$R = \frac{BPr \cap BPe}{BPr}$$

Ecuación 1 Exhaustividad

$$P = \frac{BPr \cap BPe}{BPe}$$

Ecuación 2 Precisión

Las medidas de efectividad gradadas (P_g y R_g) [82], proporcionan una clasificación de los BP_t considerados similares a un BP_q de acuerdo a diferentes niveles de relevancia. De esta manera, mientras P y R solo consideran la cantidad de elementos relevantes y recuperados, P_g y R_g tienen en cuenta la suma total de grados de relevancia entre los BP . En el presente trabajo se utilizaron las ecuaciones 3 y 4 [83] para evaluar P_g y R_g , relacionando el ordenamiento de los BP obtenidos por la herramienta automática (f_e) y el ordenamiento de las evaluaciones manuales (f_r). En estas ecuaciones se midió la efectividad de la recuperación de una herramienta al comparar una consulta BP_q con cada elemento de una colección BP_t . Por simplicidad se considera que $BP_q = Q$ y que $BP_t = T$.

$$Rg = \frac{\sum_{T_i \in T} \min\{f_r(Q, T_i), f_e(Q, T_i)\}}{\sum_{T_i \in T} f_r(Q, T_i)}$$

Ecuación 3 Exhaustividad Gradada

$$Pg = \frac{\sum_{T_i \in T} \min\{f_r(Q, T_i), f_e(Q, T_i)\}}{\sum_{T_i \in T} f_e(Q, T_i)}$$

Ecuación 4 Precisión Gradada

I.3.1 Análisis de Calidad del Repositorio

Análisis A: Estructura

	Q1	Q2	Q3	Q4	Q5	Q6
Rg	0,85	0,33	0,72	0,82	0,90	0,91
Pg	0,17	0,12	0,14	0,26	0,19	0,09
R	0,84	0,39	0,60	0,86	0,86	0,83
P	0,99	1,00	0,88	0,93	1,00	0,91

Tabla I.3-4: Precisión y Exhaustividad Estructural para cada consulta del repositorio

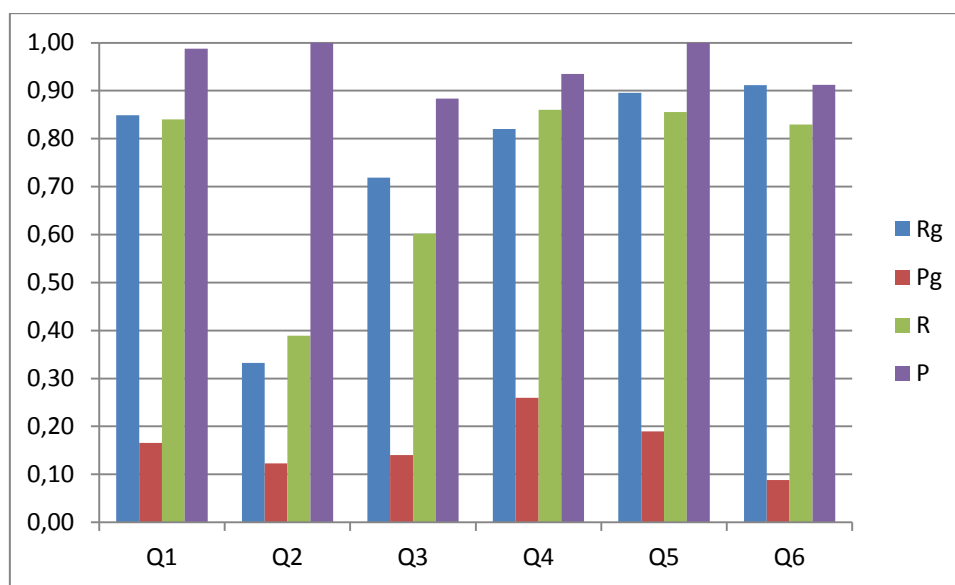


Figura I.3-1: Resultados Precisión y Exhaustividad Estructural para cada consulta del repositorio

ANÁLISIS B: Lingüística

	Q1	Q2	Q3	Q4	Q5	Q6
Rg	0,85	0,41	0,63	0,76	0,87	0,86
Pg	0,38	0,43	0,31	0,51	0,45	0,13
R	0,87	0,39	0,58	0,89	0,86	0,83
P	0,98	1,00	0,82	0,87	1,00	0,90

Tabla I.3-5: Resultados Precisión y Exhaustividad Lingüística para cada consulta del repositorio

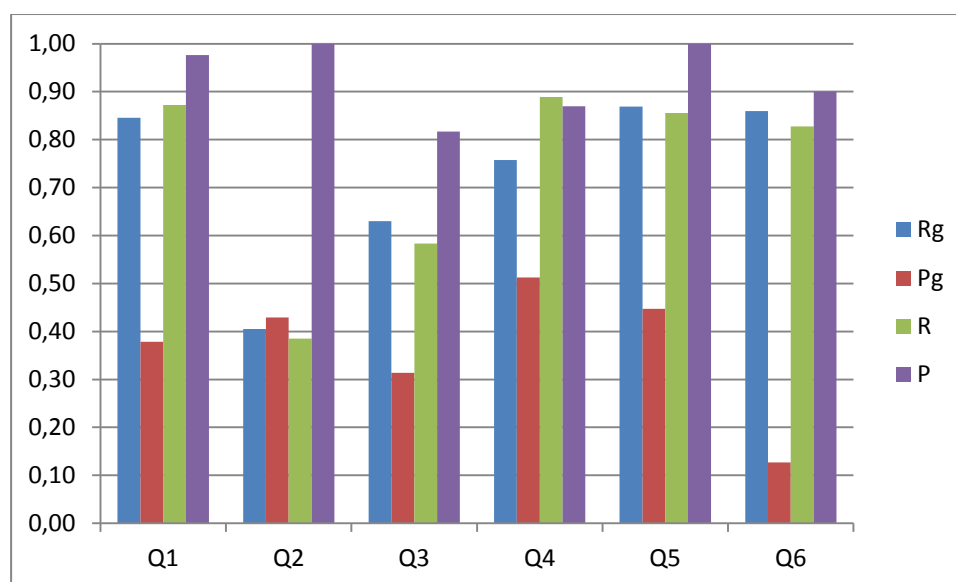


Figura I.3-2: Resultados Precisión y Exhaustividad Lingüística para cada consulta del repositorio

ANÁLISIS C: Comportamiento Secuencial

	Q1	Q2	Q3	Q4	Q5	Q6
Rg	0,89	0,20	0,87	0,86	0,09	0,95
Pg	0,20	0,10	0,23	0,26	0,97	0,12
R	0,92	0,25	0,88	0,97	0,22	0,97
P	0,82	0,41	0,77	0,67	0,81	0,74

Tabla I.3-6: Resultados Precisión y Exhaustividad para Comportamiento Secuencial para cada consulta del repositorio

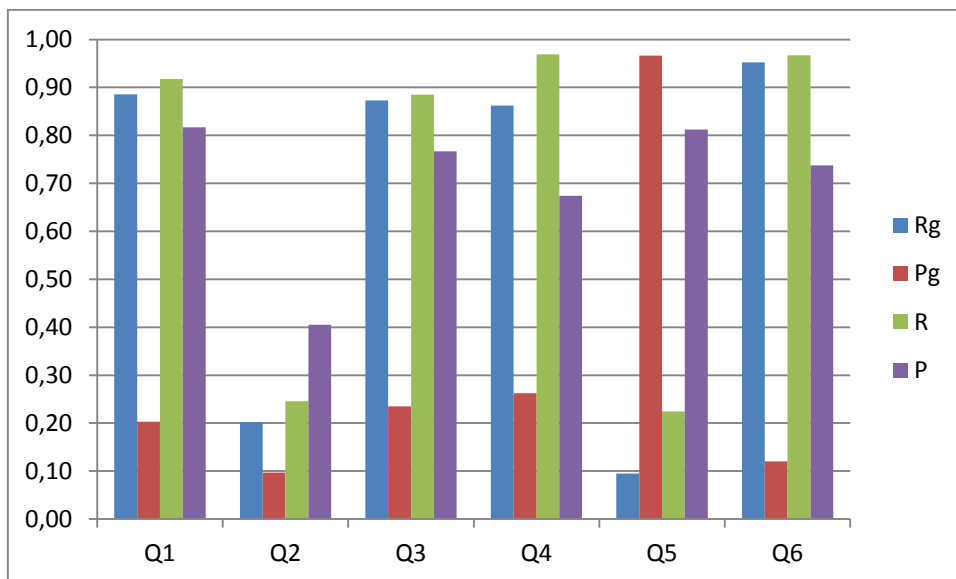


Figura I.3-3: Resultados Precisión y Exhaustividad para Comportamiento Secuencial para cada consulta del repositorio

I.3.2 Análisis de Calidad del Analizador Estructural y Semántico

ANÁLISIS B: Estructura

	Q1	Q2	Q3	Q4	Q5	Q6
Rg	0,20	0,27	0,11	0,37	0,26	0,37
Pg	0,68	0,82	0,75	0,79	0,71	0,57
R	0,12	0,26	0,05	0,20	0,16	0,11
P	0,99	1,00	1,00	0,83	1,00	1,00

Tabla I.3-7: Precisión y Exhaustividad Estructural para cada consulta del analizador estructural y semántico

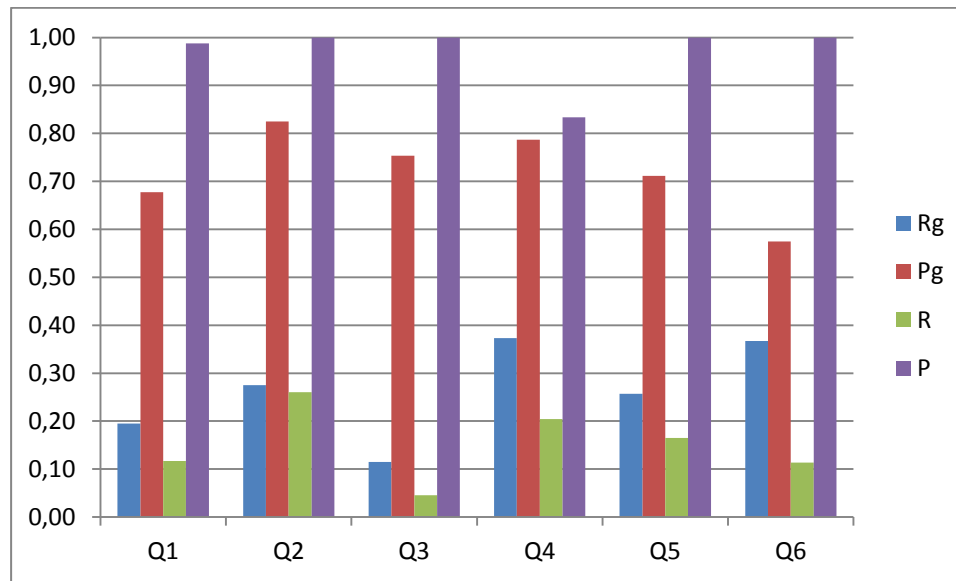


Figura I.3-4: Precisión y Exhaustividad Estructural para cada consulta del analizador estructural y semántico

ANÁLISIS B: Lingüística

	Q1	Q2	Q3	Q4	Q5	Q6
Rg	0,33	0,42	0,28	0,40	0,35	0,38
Pg	0,52	0,36	0,79	0,70	0,63	0,34
R	0,14	0,40	0,07	0,30	0,43	0,53
P	0,27	0,19	0,25	0,75	0,38	0,22

Tabla I.3-8: Precisión y Exhaustividad Lingüística para cada consulta del analizador estructural y semántico

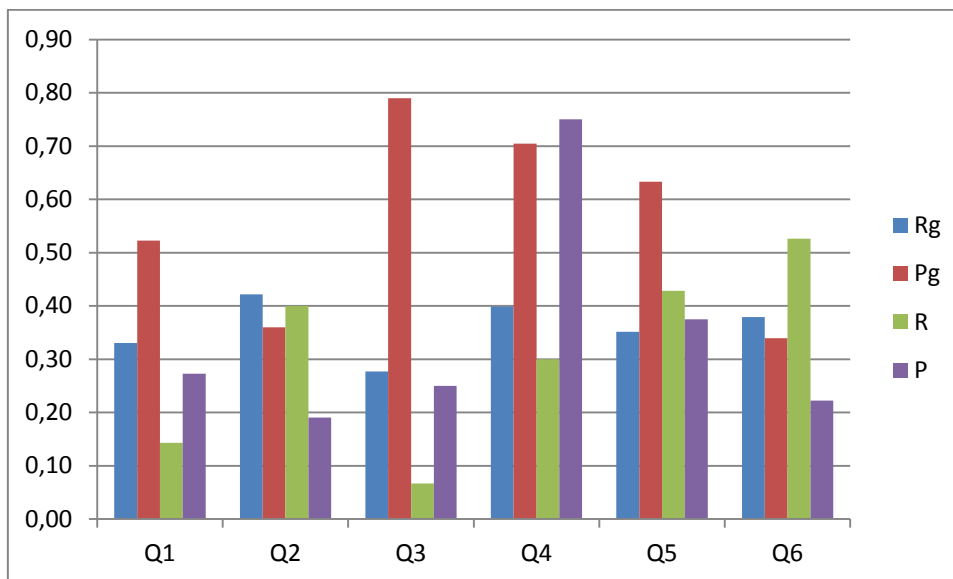


Figura I.3-5: Precisión y Exhaustividad Lingüística para cada consulta del analizador estructural y semántico

ANÁLISIS C: Comportamiento Secuencial

	Q1	Q2	Q3	Q4	Q5	Q6
Rg	0,09	0,08	0,07	0,17	0,09	0,19
Pg	1,00	0,50	0,92	0,98	0,97	1,00
R	0,15	0,13	0,06	0,31	0,22	0,16
P	1,00	0,53	0,75	0,83	0,81	1,00

Tabla I.3-9: Precisión y Exhaustividad para el Comportamiento Secuencial para cada consulta del analizador estructural y semántico

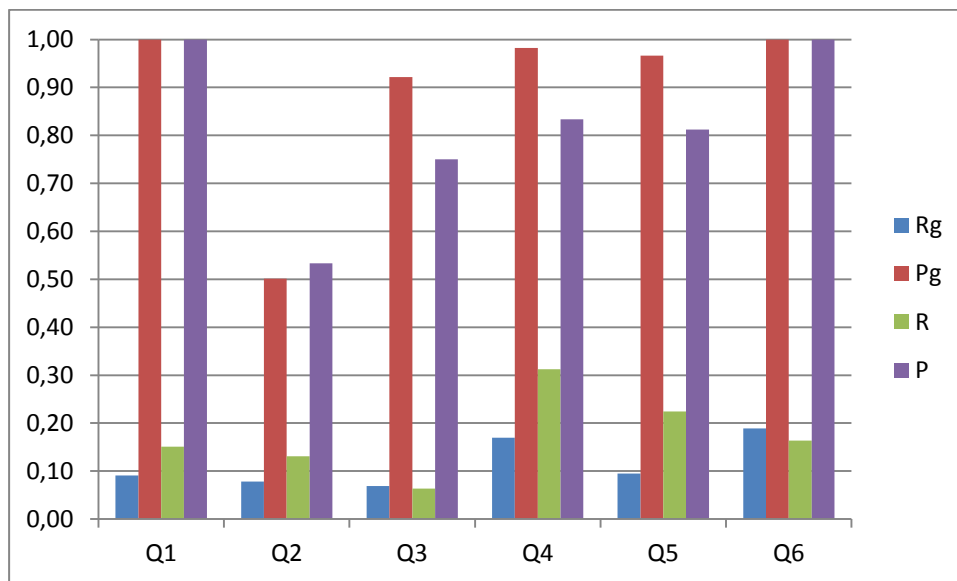


Figura I.3-6: Precisión y Exhaustividad para el Comportamiento Secuencial para cada consulta del analizador estructural y semántico

Referencias

- [1] N. Russell, *et al.*, "Workflow Control-Flow Patterns: A Revised View. ", BPM Center *BPM-06-22*, 2006.
- [2] The-WfMC, "Workflow Management Coalition Terminology & Glossary," The-WfMC, Winchester-UK1999.
- [3] A. Gomez-Perez, *et al.*, "Ontological Engineering: with examples from the areas of Knowledge Management, E-Commerce And The Semantic Web," Londres2004.
- [4] F. Manola and E. Miller, "RDF Primer," 2004.
- [5] D. Brickley and R. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema," 2004.
- [6] J. Dinos, "Arquitectura de un sistema basado en agentes para la recuperación de metadatos RDF en base a una ontología de documentos," 2004.
- [7] O. Corcho and A. Gómez-Pérez, "A Roadmap to Ontology Specification Languages," 2000.
- [8] L. Silva, "Representación de ontologías en la web semántica," 2002.
- [9] L. Hardman, *et al.*, "Hypermedia and the semantic web: a research agenda," 2002.
- [10] L. Hernández and G. Villa, "Limitaciones de la Web Actual - Futuro de la Web," 2009.
- [11] S. Bechhofer, *et al.*, "DAML+OIL is not Enough," 2001.
- [12] P. Castells, "La web semántica " 2004.
- [13] F. García, "Sistema basado en Tecnologías del Conocimiento para Entornos de Servicios Web Semánticos," 2007.
- [14] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language: Overview," 2004.
- [15] U. Keller, *et al.*, "WSMO Web Service Discovery," 2004.
- [16] J. de Bruijn, *et al.*, "The Web Service Modeling Language WSML," 2005.
- [17] H. Lausen, *et al.*, "Wsmml - A Language Framework for Semantic Web Services," 2005.
- [18] D. Fensel, *et al.*, "OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web," 2005.
- [19] C. Feier and J. Domingue, "WSMO Primer," 2005.
- [20] A. S. Marin Dimitrov, Mihail Konstantinov, "WSMO Studio Users Guide v. 1.2.7," ed, 2007.
- [21] J. Frankowski, *et al.*, "Collaborative Ontology Development in Real Telecom Environment," 2007.
- [22] R. Dawidziuk and P. Cieslak, "A proof-of-concept ontology of telecommunication services," 2007.
- [23] M. Rój, "Techniques for ontologydriven semantic interface artefacts," 2007.
- [24] S. Watkins, *et al.*, "WP 8: Case Study B2B in Telecommunications: Prototype Platform Design," 2005.

- [25] S. Wahle, "Network Domain Federation - An Architectural View on How to Federate Testbeds.," 2008.
- [26] A. Devitt, *et al.*, "Ontology-driven Automatic Construction of Bayesian Networks for Telecommunication Network Management," 2006.
- [27] TeLQAS, "Proyecto TeLQAS: Telecommunication Literature Question Answering System.."
- [28] X. Qiao and X. Li, "Semantic Web Application for Telecommunication Services," 2010.
- [29] J. Martinez and N. Pérez, "YATOSP: Marco de referencia semántico para el sector Telco," 2008.
- [30] S. Heymans, "Business Process Ontology Framework," 2007.
- [31] J. Fleck, "Overview of the Structure of the NGOSS Architecture," 2003.
- [32] *Enhanced Telecom Operations Map (eTOM)*, I. T. Map Recommendation, 2005.
- [33] R. Rodriguez and L. Garcia, "La Gestión de los Procesos de Negocio en las Empresas de Telecomunicaciones," Santa Clara, Cuba2008.
- [34] M. Burrows and M. Pendlebury, "eTOM, ITIL & ISO/IEC20000 Harmonization," Reino Unido2008.
- [35] TM-Forum. (2005). *Information Framework (SID) in Depth*. Available: <http://www.tmforum.org/InformationFramework/6647/home.html>
- [36] J. Lambea, "Business SLA Management," 2009.
- [37] NGOSS, "NGOSS Architecture Technology Neutral Specification, TMF 053," 2002.
- [38] A. Jędrzejczak, *et al.*, "Deliverable 4.5: sBPMN and sEPC to BPMP Translation," SUPER, Deliverable04/2008 2008.
- [39] N. Pérez, *et al.*, "Deliverable 9.5. Digital Asset Management Prototype," SUPER, Deliverable10/2008 2008.
- [40] N. Steinmetz and I. Toma, "WSML Language Reference," WSMO-ORG, Deliverable D16.1v1.0, 08/08/2008 2008.
- [41] R. Diestel, *Graph Theory*. New york, USA, 2000.
- [42] B. Hendrickson and R. Leland, "A multilevel algorithm for partitioning graphs," presented at the Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM), San Diego, California, United States, 1995.
- [43] W. Wilson. (2010, 17/12/2011). *Methods of Search*. Available: <http://www.cse.unsw.edu.au/~billw/Justsearch.html>
- [44] C. Okkyung and H. Sangyong, "Flexible Rule-Based Web Services System for Users," *Next Generation Web Services Practices, International Conference on*, vol. 0, pp. 1-4, 2008.
- [45] S. Zampelli, *et al.*, "Solving subgraph isomorphism problems with constraint programming," *Constraints*, vol. 15, pp. 327-353, 2010.
- [46] R. MELERO, "Acceso abierto a las publicaciones científicas: definición, recursos, copyright e impacto [En línea]," *El profesional de la información*, vol. 14, pp. 255-266, 2005.
- [47] J. P. Marta Sabou, "Towards semantically enhanced Web service repositories," 2007.
- [48] M. Maleshkova, "Acquisition and Management of Semantic Web

- Service Descriptions," ed: .NET XML Web Services Repertory, 2008.
- [49] M. S. Uwe Keller, Dieter Fensel, "WOOGLÉ meets Semantic Web Fred," 2004.
 - [50] E. A.-M. a. Q. H. Mahmoud, "Investigating Web Services on the World Wide Web," *www 2008*, 2008.
 - [51] e. a. Karl Czajkowski, "The WS-Resource Framework," 2004.
 - [52] S. F. a. S. L. Julie Allinson, "SWORD: Simple Web-service Offering Repository Deposit," 2009.
 - [53] H. L. Daniel Bachlechner, Katharina Siorpaes, "Web Service Discovery – A Reality Check," 2006.
 - [54] D. Bianchini, *et al.*, "Evaluating Similarity and Difference in Service Matchmaking," presented at the INTEROP-EMOI'06, Luxembourg, 2006.
 - [55] Z. Yan, *et al.*, "Business Process Model Repositories - Framework and Survey," 2009.
 - [56] M. K. M. KACZMAREK, "Deliverable 3.4. Business Process Library Final Prototype," ed, 2008.
 - [57] M. K. KACZMAREK, Mihail, "Deliverable 3.1 Business Process Library Design and first Prototype," ed, 2007.
 - [58] B. Barry, "Deliverable 6.9 Process Ontology and Query Reasoner - Final Implementation," 2008.
 - [59] H. Al-mubaid and H. A. Nguyen, "A Cluster-Based Approach for Semantic Similarity in the Biomedical Domain," in *In Proc. The 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, ed, 2006.
 - [60] M. W. ZAREMBA, Brian; BROWNE, Jimmie;, "ebXML Registry/Repository Implementation in the Agri-food industry," ed, 2003.
 - [61] P. Resnik, "Selection and Information: A Class-Based Approach to Lexical Relationships," University of Pennsylvania 1993.
 - [62] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," presented at the Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1, Montreal, Quebec, Canada, 1995.
 - [63] J. VANHATALO, "Building and Querying a Repository of BPEL Process Specifications. Research Report," Master Science Degree, Université Nice Sophia Antipolis, 2004.
 - [64] J. VANHATALO, "Building and Querying a Repository of BPEL Process Specifications," ed. Zurich Laboratory: IBM Research, 2004.
 - [65] K. K. I. Choi, and M. Jang., "An xml-based process repository and process query language for integrated process management," *Knowledge and Process Management*, pp. 303-316, 2007.
 - [66] H. J. I. Choi, M. Song, and Y. Eyu., "Ipm-epdl: an xml-based executable process definition language," *Computers in Industry*, vol. 56, pp. 85-104, 2005.
 - [67] J. A. M. M. Song, and I.B. Arpinar., "RepoX: An XML Repository for Workflow Designs and Specifications," University of Georgia, USA 2001.
 - [68] J. Z. T. Theling, P. Loos, and D. Vanderhaeghen, "An architecture for collaborative scenarios applying a common bpmn-repository.," *In Proceedings of DAIS 2005*, pp. 169-180, 2005.

- [69] H. O. G. Decker, and M. Weske., "Oryx C An Open Modeling Platform for the BPM Community," *In Proceedings of BPM 2008*, pp. 382-385, 2008.
- [70] A. E. C. Beerli, T. Milo, and A. Pilberg., "Bp-mon: Query-based monitoring of bpm business processes. ," *SIGMOD Record* vol. 1, pp. 21-24, 2008.
- [71] K. C. T.W. Malone, and G.A. Herman, "Organizing Business Knowledge: The MIT Process Handbook.," *MIT Press*, 2003.
- [72] J. L. S. Fiorini, and C. Lucena., "Process reuse architecture.," presented at the *In Proceedings of CAiSE 2001*, Interlaken, Switzerland, 2001.
- [73] G. Yang., "Towards a library for process programming," presented at the *In Proceedings of BPM 2003*, Eindhoven, The Netherlands, 2003.
- [74] X. L. C. Liu, X. Zhou, and M. Orlowska, "Building a repository for workflow systems.," in *In Proceedings of Technology of Object-Oriented Languages and Systems*, Nanjing, China, 1999, pp. 348-357.
- [75] M. L. A. Wasser, and R. Karni., "Processgene query- a tool for querying the content layer of business process models.," in *Proceedings of BPM 2006*, Vienna, Austria., 2006, pp. 1-8.
- [76] S. S. R. Lu, and G. Governatori., "On managing business processes variants.," *Data & Knowledge Engineering*, vol. 68, pp. 642-664, 2009.
- [77] A. P. I. Markovic, and N. Stojanovic., "A framework for queying in business process modeling," *Multikonferenz Wirtschaftsinformatik*, 2008.
- [78] a. M. B. M. Elhadad, "Effective business process outsourcing: The prosero approach," *International Journal of Interoperability in Business Information Systems*, vol. 3, 2008.
- [79] M. a. R. La Rosa, Hajo A. and Aalst, Wil M.P. van der and Dijkman, Remco M. and Mendling, Jan and Dumas, Marlon and Garcia-Banuelos, Luciano, "APROMORE : An Advanced Process Model Repository," Queensland University of Technology 2009.
- [80] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*: McGraw-Hill, Inc., 1986.
- [81] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [82] U. Küster and B. König-Ries, "On the Empirical Evaluation of Semantic Web Service Approaches: Towards Common SWS Test Collections," presented at the *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, 2008.
- [83] D. A. Buell and D. H. Kraft, "Performance measurement in a fuzzy retrieval environment," *SIGIR Forum*, vol. 16, pp. 56-62, 1981.